

FM07: Trust Accounts

November 1, 2004

Table Of Contents

| | |
|---|----|
| 1.1. INTRODUCTION | 4 |
| 1.2. PAGES | 6 |
| 1.2.1. <i>Page-Trust Account</i> | 6 |
| 1.2.1.1. Page Overview | 6 |
| 1.2.1.2. Page Information | 7 |
| 1.2.1.3. Background Processing | 8 |
| 1.2.1.4. Save Processing | 9 |
| 1.2.1.5. CRUD Matrix | 9 |
| 1.2.2. <i>Tab-Benefit Record</i> | 10 |
| 1.2.2.1. Page Overview | 10 |
| 1.2.2.2. Page Information | 11 |
| 1.2.2.3. Background Processing | 13 |
| 1.2.2.4. Save Processing | 14 |
| 1.2.2.5. CRUD Matrix | 14 |
| 1.2.3. <i>Page – Maintain Benefit Record</i> | 15 |
| 1.2.3.1. Page Overview | 15 |
| 1.2.3.2. Page Information | 16 |
| 1.2.3.3. Background Processing | 18 |
| 1.2.3.4. Save Processing | 19 |
| 1.2.3.5. CRUD Matrix | 19 |
| 1.2.4. <i>Tab- Ledger</i> | 20 |
| 1.2.4.1. Page Overview | 20 |
| 1.2.4.2. Page Information | 21 |
| 1.2.4.3. Background Processing | 25 |
| 1.2.4.4. Save Processing | 26 |
| 1.2.4.5. CRUD Matrix | 29 |
| 1.2.5. <i>Page- Maintain Ledger Record</i> | 31 |
| 1.2.5.1. Page Overview | 31 |
| 1.2.5.2. Page Information | 31 |
| 1.2.5.3. Background Processing | 33 |
| 1.2.5.4. Save Processing | 33 |
| 1.2.5.5. CRUD Matrix | 33 |
| 1.2.6. <i>Page - Bank</i> | 34 |
| 1.2.6.1. Page Overview | 35 |
| 1.2.6.2. Page Information | 35 |
| 1.2.6.3. Background Processing | 37 |
| 1.2.6.4. Save Processing | 37 |
| 1.2.6.5. CRUD Matrix | 38 |
| 1.2.7. <i>Pop-up - Interest</i> | 39 |
| 1.2.7.1. Page Overview | 39 |
| 1.2.7.2. Page Information | 40 |
| 1.2.7.3. Background Processing | 41 |
| 1.2.7.4. Save Processing | 41 |
| 1.2.7.5. CRUD Matrix | 42 |
| 1.2.8. <i>Page – Maximum Balance</i> | 43 |
| 1.2.8.1. Page Overview | 43 |
| 1.2.8.2. Page Information | 44 |
| 1.2.8.3. Background Processing | 44 |
| 1.2.8.4. Save Processing | 45 |
| 1.2.8.5. CRUD Matrix | 45 |
| 1.2.9. <i>Page – Closed Cases Trust Account Search Page</i> | 46 |
| 1.2.9.1. Page Overview | 46 |
| 1.2.9.2. Page Information | 47 |
| 1.2.9.3. Background Processing | 47 |

| | |
|--|-----|
| 1.2.9.4. Save Processing..... | 48 |
| 1.2.9.5. CRUD Matrix..... | 49 |
| 1.2.10. Page – Non-WiSACWIS Benefits Page..... | 50 |
| 1.2.10.1. Page Overview..... | 50 |
| 1.2.10.2. Page Information..... | 50 |
| Service Category..... | 51 |
| Service Type..... | 51 |
| Facility Name..... | 51 |
| Provider ID..... | 51 |
| Maint. Amt..... | 51 |
| Title XIX Amt..... | 51 |
| 1.2.10.3. Background Processing..... | 52 |
| 1.2.10.4. Save Processing..... | 52 |
| 1.2.10.5. CRUD Matrix..... | 52 |
| 1.2.11. Page – Maintain Non-WiSACWIS Benefits Page..... | 53 |
| 1.2.11.1. Page Overview..... | 53 |
| 1.2.11.2. Page Information..... | 53 |
| 1.2.11.3. CRUD Matrix..... | 57 |
| 1.2.12. Table Descriptions..... | 58 |
| 1.2.13. Reference Data..... | 60 |
| 1.2.13.1. Drop Downs..... | 60 |
| 1.2.13.2. List Boxes..... | 60 |
| 1.2.14. Automated Messages..... | 60 |
| 1.2.15. Checklists..... | 60 |
| 1.2.16. Ticklers..... | 60 |
| 1.2.16.1. Spend Down Needed for FFP child..... | 60 |
| 1.2.16.2. Spend Down Needed for Non-FFP child..... | 61 |
| 1.2.17. Notifications..... | 62 |
| 1.2.18. Text Templates..... | 63 |
| 1.2.19. Reports..... | 63 |
| 1.2.19.1. FM07-01 Trust Account Current Balance Report..... | 63 |
| 1.2.19.2. FM07-02 Trust Account Reconciliation Report..... | 63 |
| 1.2.19.3. FM07-03 Placement Ending Report..... | 63 |
| 1.2.19.4. FM07-04 Deposit History Report..... | 63 |
| 1.2.19.5. FM07-05 Trust Account Balances for Closed Cases..... | 63 |
| 1.2.19.6. FM07-06 Trust Account Drawdowns..... | 63 |
| 1.2.20. Triggers..... | 63 |
| 1.2.20.1. Online Trigger Txn Code 100..... | 63 |
| 1.2.20.2. Online Trigger Txn Code 101..... | 63 |
| 1.2.20.3. Online Trigger Txn Code 103..... | 64 |
| 1.2.20.4. Online Trigger Txn Code 105..... | 64 |
| 1.2.20.5. Online Trigger Txn Code 106..... | 65 |
| 1.3. BATCH PROGRAMS..... | 66 |
| 1.3.1. Spread Interest Batch..... | 66 |
| 1.3.2. Trust Account Deposit Batch..... | 73 |
| 1.3.3. KIDS to WiSACWIS Daily Batch..... | 80 |
| 1.4.4 WiSACWIS to KIDS Daily Batch..... | 97 |
| 1.4.5 WiSACWIS to KIDS Monthly Batch..... | 125 |
| 1.4.6 EFT Ledger Update batch..... | 129 |
| 1.4.7 KIDS Financial Ledger Update batch..... | 136 |
| 1.4.8 Trust Account Spend Down Tickler Update..... | 143 |
| 1.4. ONLINE COMPONENTS..... | 145 |
| 1.4.1. User Interface Components..... | 145 |

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

1.1. Introduction

FM07: Trust Account topic paper covers the creation and maintenance of a Trust Account on behalf of children wards of State. Only the Trust Account Manager will have access to maintain benefit money on behalf of these children. Each child may only have two trust accounts opened at a single time per county. The Dedicated Trust Account will hold SSI money that is received in a single payment in excess of six months. The General Trust Account will hold all other benefits. These two accounts will be established as separate interest bearing bank accounts. The break down of each child's portion, including all withdrawals and deposits, will be maintained in the Ledger tab of the Trust Account page.

Maximum balance levels can be established for trust accounts. Using the Maximum Balance page, the Trust Account Manager can establish a maximum balance for Federal Financial Participation (FFP) and Non-FFP children. When the child's balance reaches a specified percentage of this balance, a tickler will be sent to the trust account manager to initiate a spend-down. Once the trust account manager has confirmed that money can be spent, they will contact the primary worker.

Interest earned will be maintained through the Interest page of this module. The interest amount earned in each bank account can be entered in this page. A spread interest batch will pick up this amount and prorate each child's portion. This will indicate a deposit entry on the ledger record.

Caseworkers and other assignees of the Case will have the only view access to the Trust Account page. This module is integrated with the Trust Account Draw Down batch (See *FM04a Reimbursement*), which uses benefit money towards the cost of ongoing care for a child. The Trust Account Draw Down batch will withdraw available money from the General Account. The Trust Account Manager will have the ability to indicate which benefits are available for ongoing care. The Trust Account Manager must also write a check to the State once the batch has been determined how much money is reimbursable to the State for that period.

Child Support money is also available to the State to reimburse the cost of ongoing care. The referral process to the KIDS (Kids Information Data System) of Department of Workforce Development is described in this paper. Two batch programs send referrals and financial information to the KIDS System. The KIDS system sends back both child support amounts for each child and any changes in demographic information on the child or parents.

An interface with Firststar Bank will download and post to the trust account ledger all EFT transactions except Child Support deposits. The deposits that are recorded automatically are transactions from SSI-State, SSI-Federal, SSA-Survivor's, SSA-Disability, VA and Railroad Benefits.

The Closed Cases Trust Account Search page allows workers to search for general trust accounts in the database for closed cases. This page also allows workers to

create a general trust account if one does not exist. Workers will not be able to create a new general trust account if one already exists for that case, child, and county.

The Non-WiSACWIS Benefits page allows workers to record and delete trust funds received for non-WiSACWIS children. This information could then be reported to BFS. The only money recorded on this page is for children who are known to be FFP.

1.2. Pages

1.2.1. Page-Trust Account

Person/Worker Information

Name: Ashley Ash SSN Child: 198-76-5432 Trust Acct #: General **Close Acct**

Case: Anna Ash SSN Case: 798-46-8432 Closed Date: 00/00/0000 Dedicated

Worker: Rob Thomas Phone: (603)673-2846 Bank: Firstar - Milwaukee - General 12121212

Benefit Record Ledger

Trust Account Benefit Record

| View | Type | Claim # | Appl. Date | Rjct. Date | Effct. Date | Amount | Closing Date |
|---------------|------|---------|------------|------------|-------------|--------|--------------|
| Insert | | | | | | | |

Benefit Record Details

Type: Rejection Date: Closing Date:

Claim Number: Effective Date: Personal Needs Deduction: Periodic Days: EFT Deposit:

Application Date: Amount:

Benefit is available for cost of care expenses

Description:

Options: **Go** **Save** **Close**

1.2.1.1. Page Overview

Navigation

The Trust Account page is created after the worker selects Trust Account General or Trust Account Dedicated from the Eligibility drop down box and a case and case participant from the Create Case Work page. After the worker selects the Create button, the Trust Account page is displayed. To open an existing trust account, a worker can select the Eligibility icon under a case folder in the Cases expando. When selecting the hyperlink for the trust account instance, the record opens with the appropriate information viewable for the worker.

Page Summary

Each child can have one General Trust Account and one Dedicated Trust Account open at a single time per county. The Dedicated Trust Account will be used to hold

SSI payments that are received in single payment for a period greater than six months. The General Trust Account will hold all other benefits and will be the account from which the State will reimburse itself for the cost of ongoing care (See *FM04a: Reimbursement*; Trust Account Draw Down batch). When dedicate funds have been approved for State reimbursement, the trust account manager must manually transfer funds from the Dedicated Account to the General Account. Once the page is created, the Bank field will pre-fill to the bank associated with the worker's County, Trust Account type created, and will not be editable. The final feature of this header is the Closed Date field. When the trust account manager is prepared to close this trust account, the balance and outstanding cost of care must be zeroed out and the Closed Account button will become enabled. When the worker selects the Closed Account button, the Closed Date is populated and the page is saved, the trust account will only be view accessible. A new trust account can now be opened or the existing closed trust account can be re-opened.

1.2.1.2. Page Information

Box: Person/Worker Information

Fields:

| | |
|----------------|---|
| Name: | First and Last Name of the child for whom this trust account is established. System derived; Not user editable. Retrieved using TRUST_ACCOUNT.id_prsn from PERSON.nm_frst, nm_lst. |
| Case: | Name of the case in which this trust account resides. System derived; Not user editable. Retrieved from CASE.nm_case_lst, nm_case_frst, nm_case_mdl_intl. |
| Worker: | Name of the primary worker. System derived; Not user editable. Retrieved from PERSON_nm_frst, nm_lst, where PERSON.id_prsn equals ASSIGNMENT.id_grp_lvl1 = id_case and cd_asgn_ctry = 1 and cd_asgn_role = 1. |
| SSN Child: | Social Security Number of the child as stored in Person Management. System derived; Not user editable. Retrieved from PERSON.id_ssn. |
| SSN Case: | Social Security Number of the Reference Person. System derived; Not user editable. Retrieved from PERSON.id_ssn where PERSON.id_prsn = CASE.id_pns_ref |
| Phone: | Phone number of the primary worker assigned to case. System derived; Not user editable. Retrieved from WORKER.ad_work_phone. |
| Trust Acct. #: | The unique identifier associated with the trust account. System derived; Not user editable. Retrieved from |

- TRUST_ACCOUNT.id_trust_acct after record has been initially saved.
- Closed Date:** The date the trust account is ended for a child. Once this field is populated and the page saved, this field will become disabled as well as all other fields in the Trust Account page for this child. This Trust Account will be view only access. A new Trust Account for this child and account can now be created. User entered date field; User editable when the Close Acct. Button is selected; Required; Defaults to 00/00/0000. Maps to TRUST_ACCOUNT.dt_closing.
- Bank:** Displays the Bank Name, County and Account number of the banks created in the Bank page. System derived; Not user editable. Filter BANK.cd_cnty to equal WORKER.cd_ofc_div.
- Bank Type:** Identifies the type of bank account associated with the trust account record (General or Dedicated). System derived; Not user editable. Filter BANK.cd_cnty to equal WORKER.cd_ofc_div.
- Buttons:**
- Close Acct.:** Enabled only when total balance and outstanding cost of care for this id_trust_acct is zero. Enables the Closed Date field. On save, the Trust Account page becomes disabled.
 - Open Acct.:** Enabled only when total balance for this id_trust_acct is zero. Selecting this button will re-open the trust account if no other account exists for this child, case, and county. Disables the Closed Date field. On save, the Trust Account page becomes enabled.

1.2.1.3. Background Processing

1. When the total balance and outstanding cost of care for this id_trust_acct is zero, the “Closed Date” field will become enabled.
2. When opening the trust account from the outliner and TRUST_ACCOUNT.dt_closing <> null, then all fields on the Benefit Record tab and Ledger tab will be frozen. The scroll bar on the Benefit Record tab and Ledger tab will be enable for in order to review history. The Save As button will be enabled to download transaction ledger listings on the Ledger Tab and the Open Acct. button will enabled.

3. When the Close Acct. button is selected, the user is prompted with a message stating “This will close the account. Do you wish to continue?” If the worker selects ‘Yes’, the Closed Date field becomes enabled and required. Additionally, the Trust Account page disables.
4. When the Open Acct. button is selected, the user is prompted with a message stating “This Trust Account is about to be re-opened. Do you wish to continue?” If the worker selects ‘Yes’, the Closed Date field disables. Additionally, the Trust Account page enables.

1.2.1.4. Save Processing

1. Save Processing is initiated by clicking on the Save command button. Save processing is also initiated when selecting the Close button, and then answering ‘Yes’ when the system asks the user if they would like to ‘Save changes before closing the page.’
2. When a trust account page is initially saved, the Trust Acct. # field is filled with a unique identifier from the TRUST_ACCOUNT table.

1.2.1.5. CRUD Matrix

| Table Name | CRUD |
|-------------------|-------------|
| TRUST_ACCOUNT | CRU |
| PERSON | R |
| WORKER | R |
| CASE | R |
| ASSIGNMENT | R |
| BANK | R |
| CODE_DESC | R |

1.2.2. Tab-Benefit Record

Person/Worker Information

Name: Annabel Abby SSN Child: 789-79-7955 Trust Acct #: 9221006 General
Case: Amy Abby SSN Case: 353-41-5655 Closed Date: 00/00/0000 Dedicated
Worker: Frank Fox Phone: Bank: Firststar - Milwaukee - General 12121212

Benefit Record Ledger

Trust Account Benefit Record

| View | Type | Claim # | Appl. Date | Rjct. Date | Effct. Date | Amount | Closing Date | |
|----------------------------------|---------------|-------------|------------|------------|-------------|--------|--------------|---|
| <input checked="" type="radio"/> | Child Support | 125-46-5213 | 00/00/0000 | 00/00/0000 | 08/19/2003 | \$0.00 | 00/00/0000 | Edit Delete |

Benefit Record Details

Type: Child Support Rejection Date: 00/00/0000 Closing Date: 00/00/0000
Claim Number: 125-46-5213 Effective Date: 08/19/2003 Personal Needs Deduction: Periodic Days: EFT Deposit:
Application Date: 00/00/0000 Amount: \$0.00 \$0.00 No
Benefit is available for All Outstanding cost of care expenses
Description:

Options:

1.2.2.1. Page Overview

The Benefit Record tab will keep a history and show the current status of each of the benefits a child is expected to receive. This tab provides the trust account manager with the ability to record when a benefit application was submitted, and when benefit payments are approved or rejected and then finally closed. A new record is created through the Insert button in the Trust Account Benefit Record group box. When selected, the Maintain Benefit Record page opens for the worker to enter benefit information. After the user creates a benefit record and presses Continue on the Maintain Benefit Record page, the newly created record appears in the Benefit Record Details group box on the Benefit Record tab. To update or delete an already existing benefit record, the worker can select the Edit or Delete hyperlinks located to the right of each record. The Maintain Benefit Record page is discussed in the following section of this document.

To view details of a benefit record, the radio button located to the left of each benefit record is selected. When selected, the associated information appears in the Benefit Record Details group box.

The effective date is the date after which benefit received can be used towards ongoing care process (See *FM04a Reimbursement, Trust Account Draw Down Batch*).

Any payments with a begin date prior to this effective date will not be used to reimburse the State.

1.2.2.2. Page Information

Box: Trust Account Benefit Record

Fields:

| | |
|---------------|--|
| View: | This column has radio buttons available for selection by a worker; When selected, the details from the benefit record are filled into the Benefit Record Details group box; User selected radio buttons; User editable; Defaults to the first record in the repeating group. |
| Type: | This column displays the benefit type. System derived; Not user editable. Retrieved from BENEFIT.cd_bnft_typ. |
| Claim #: | Claim number associated with this benefit. System derived; Not user editable. Retrieved from the BENEFIT.id_clm. |
| Appl. Date: | Date this benefit with this claim number was applied for obtaining this source of funding. System derived; Not user editable. Retrieved from BENEFIT.applctn. |
| Rjct. Date: | Date the benefit with this claim number might have been rejected by funding source. System derived; Not user editable. Retrieved from BENEFIT.dt_rjct. |
| Effct. Date: | Date the benefit with this claim number was made effective and is the date from which benefits can be used towards ongoing care. Benefit money recording in the ledger prior to this date will not be withdrawn by Trust Account Draw Down batch (See <i>FM04a: Reimbursement</i>). System derived; Not user editable. Retrieved from PAYMENT.dt_efctv. |
| Amount: | The dollar amount of this benefit type that is expected to be received from this benefit funding source for this claim number. System derived from the Benefit Record page; Not user editable. Retrieved from BENEFIT.am_bnft. |
| Closing Date: | Date the benefit is no longer to be expected to be received from this benefit funding source for this claim number. System derived; Not user editable. Retrieved from BENEFIT.dt_clsng. |

Box: Benefit Record Details

- Fields:**
- Type: This field displays the benefit type. System derived; Not user editable; AFCARS field. Retrieved from BENEFIT.cd_bnft_typ.
 - Claim Number: Claim number associated with this benefit. System derived; Not user editable. Retrieved from the BENEFIT.id_clm.
 - Application Date: Date this benefit with this claim number was applied for obtaining this source of funding. System derived; Not user editable. Retrieved from BENEFIT.applctn.
 - Rejection Date: Date the benefit with this claim number might have been rejected by funding source. System derived; Not user editable. Retrieved from BENEFIT.dt_rjct.
 - Effective Date: Date the benefit with this claim number was made effective and is the date from which benefits can be used towards ongoing care. Benefit money recording in the ledger prior to this date will not be withdrawn by Trust Account Draw Down batch (See *FM04a: Reimbursement*). System derived; Not user editable. Retrieved from PAYMENT.dt_efctv.
 - Amount: The dollar amount of this benefit type that is expected to be received from this benefit funding source for this claim number. System derived; Not user editable. Retrieved from BENEFIT.am_bnft.
 - Closing Date: Date the benefit is no longer to be expected to be received from this benefit funding source for this claim number. System derived; Not user editable. Retrieved from BENEFIT.dt_clsng.
 - Personal Needs Deduction: Amount that is to be stripped off each benefit deposit and allocated to a personal needs fund that is not touched by the reimbursement batch. This amount will be stripped off by the EFT batch and the Deposit batch. System derived; Not user editable. Retrieved from BENEFIT.am_prsnl_nds.
 - Periodic Days: Number of days this benefit should be automatically uploaded to the Ledger. If equal to zero, the benefit type will not be automatically uploaded to the ledger. System

derived; Not user editable. Retrieved from BENEFIT.am_days.

EFT Deposit : This check box indicates this benefit type is expected to be automatically uploaded to the ledger through the EFT batch process. System derived; Not user editable. Retrieved from BENEFIT.cd_eft.

Benefit is available for <value> cost of care expenses:

This field indicates what payment period this benefit is available for State reimbursement. Values are 1) All Outstanding 2) Current Month 3) Prior Month 4) No. Default value will be 4) All Outstanding. System derived; Not user editable. Retrieved from BENEFIT.cd_expns.

Description: This field is used to store remarks associated with a benefit record. System derived; Not user editable. Retrieved from BENEFIT.tx_desc.

| | | |
|-----------------|---------|--|
| Links: | Edit: | Opens the Maintain Benefit Record page for the selected benefit record. |
| | Delete: | Deletes the selected benefit record from the Trust Account page. |
| Buttons: | Insert | Opens the Maintain Benefit Record page for the user to add a new benefit record. Enabled when the Trust Account is open. |
| | Save | Standard Save Processing |
| | Close | Standard Close Processing |

1.2.2.3. Background Processing

1. When the Delete hyperlink is selected for a benefit record, the user is prompted with a notification message that states “Are you sure you want to delete currently selected information?” If the user selects ‘Yes’, the record is removed from the Trust Account Benefit Record group box.
2. If the Trust Account record is closed, the Insert button is disabled. Additionally, the Edit and Delete links are not available for benefit records.
3. When a radio button is selected for a record in the Trust Account Benefit Record group box, the appropriate information is visible in the Benefit Record Details group box.

1.2.2.4. Save Processing

1. Save Processing is initiated by clicking on the Save command button. Save processing is also initiated when selecting the Close button, and then answering 'Yes' when the system asks the user if they would like to 'Save changes before closing the page.'

1.2.2.5. CRUD Matrix

| Table Name | CRUD |
|-------------------|-------------|
| TRUST_ACCOUNT | R |
| BENEFIT | CRU |

1.2.3. Page – Maintain Benefit Record

Maintain Benefit Record

Type: [Dropdown] Rejection Date: [00/00/0000] Closing Date: [00/00/0000]

Claim Number: [- -] Effective Date: [08/01/2003] Personal Needs Deduction: [] Periodic Days: [] EFT Deposit: []

Application Date: [00/00/0000] Amount: [\$0.00] [\$0.00] [] []

Benefit is available for [Dropdown] cost of care expenses

Description: [Text Field]

[Continue] [Close]

1.2.3.1. Page Overview

Navigation

This page is accessed by selecting the Insert button on the Benefit Record tab or by selecting an Edit hyperlink to the right of a benefit record.

Page Summary

The Maintain Benefit Record page is used to enter new benefit records to the Benefit Record tab. Additionally, a user can update benefit records on this page. When the user selects the Continue button, the Maintain Benefit Record page will close and the new or updated information will be passed back to the Benefit Record tab. This information will be saved to the database when the Save button is selected on the Trust Account page. However, business logic errors will be validated when the user selects the Continue button.

1.2.3.1.1. Personal Needs Allocation

The Maintain Benefit Record page also gives the trust account manager the ability to strip off a specified dollar amount off each deposited benefit. This money is held in a

personal needs fund on the ledger. This personal needs deduction is made whenever a deposit is made to the ledger via the EFT Update Ledger batch or the Deposit batch. Money allocated to the personal needs category will not be used in reimbursement process (See *FM04a Reimbursement, Trust Account Draw Down Batch*).

1.2.3.1.2. Automatic Deposit Method

The EFT check box indicates that this benefit type is expected to be uploaded to the ledger tab automatically through the EFT batch process. If an EFT check box is not selected, then the periodic days field is enabled and allows the user to indicate number of days a new ledger entry should appear with the benefit amount listed on the Benefit Record tab. If the periodic days is zero, then benefit amount will not appear on the Ledger tab, but will just remain on the Benefit Record tab. In this case the trust account manager must update the ledger manually when a deposit is made.

1.2.3.1.3. Remarks

A description field allows the trust account manager to include remarks related to each benefit type.

1.2.3.2. Page Information

Box: Maintain Benefit Record

Fields:

| | |
|-------------------|--|
| Type: | User must select a benefit type from drop down list. User selected drop down box; User editable; Required; No default value; AFCARS field. Maps to BENEFIT.cd_bnft_typ where code value is found on CODE_DESC_FINANCIAL.id_grp = BENFTYPE. |
| Claim Number: | Once a claim number has been established, this number can be stored in this field. This claim number must be a unique number within all the open Trust Accounts in eWiSACWIS. The EFT batch will be able to update the ledger using this number. User entered number field; User editable; Not required; No default value. Maps to the BENEFIT.id_clm. |
| Application Date: | Date this benefit with this claim number was applied for obtaining this source of funding. User entered date field; User editable; Not required; Defaults to 00/00/0000. Maps to BENEFIT.applctn. |
| Rejection Date: | Date the benefit with this claim number might have been rejected by funding source. User entered date field; User editable; Not required; Defaults to 00/00/0000. Maps to BENEFIT.dt_rjct. |

- Effective Date: Date the benefit with this claim number was made effective and is the date from which benefits can be used towards ongoing care. Benefit money recording in the ledger prior to this date will not be withdrawn by Trust Account Draw Down batch (See *FM04a: Reimbursement*). User entered date field; User editable; Not required; Defaults to the begin date of the child's latest initial out of home placement. Maps to BENEFIT.dt_efctv.
- Amount: The dollar amount of this benefit type that is expected to be received from this benefit funding source for this claim number. User entered number field; User editable; Not required; No default value. Maps to BENEFIT.am_bnft.
- Closing Date: Date the benefit is no longer to be expected to be received from this benefit funding source for this claim number. User entered date field; User editable; Not required; Defaults to 00/00/0000. Maps to BENEFIT.dt_clsng.
- Personal Needs Deduction: Amount that is to be stripped off each benefit deposit and allocated to a personal needs fund that is not touched by the reimbursement batch. This amount will be stripped off by the EFT batch and the Deposit batch. User entered number field; User editable; Not required; No default value. Maps to BENEFIT.am_prsnl_nds.
- Periodic Days: Number of days this benefit should be automatically uploaded to the Ledger. If equal to zero, the benefit type will not be automatically be uploaded to the ledger. This field is only enabled if EFT Deposit is not selected. User entered number field; User editable; Not required; No default value. Maps to BENEFIT.am_days.
- EFT Deposit : This check box indicates this benefit type is expected to be automatically uploaded to the ledger through the EFT batch process. When selected, the periodic days field will be disabled. User selected checkbox; User editable; Not required; Defaults to not checked. Maps to BENEFIT.cd_eft.
- Benefit is available for <value> cost of care

| | |
|-----------------|--|
| expenses: | This drop down indicates what payment period this benefit is available for State reimbursement. Values are 1) All Outstanding 2) Current Month 3) Prior Month 4) No. User selected drop down box; User editable; Required; No default value. Maps to BENEFIT.cd_expns. |
| Description: | Allows a user to store a remark for each benefit record; User entered text field; User editable; Not required; No default value. Maps to BENEFIT.tx_desc. |
| Buttons: | |
| Continue | Runs business logic checks for the benefit record. If there no are business errors, the user is navigated back to the Trust Account page. The newly entered information is passed back to the Trust Account page, Benefit Record tab. |
| Close | Standard Close Processing |

1.2.3.3. Background Processing

1. If BENEFIT.cd_eft = 1 or 2, then display box as checked. If BENEFIT.cd_eft = 0, then display box as unchecked.
2. If BENEFIT.cd_eft = 1 or 2, then disable Periodic Days fields. If BENEFIT.cd_eft = 0, then enable Periodic Days field.
3. The Amount field only accepts positive values. An edit mask prevents user from entering a negative value.
4. The Personal Needs Deduction field only accepts positive values. An edit mask prevents user from entering a negative value.
5. The Type and Benefit is Available for... fields are required and cannot have a null value at time of selecting the Continue button. If either of these fields are null, the user will receive an error validation stating "Please enter data in the fields which are highlighted - <field name>."
6. If Closing Date is less than Effective Date, the user receives an error message after pressing the Continue button stating "Close date must be greater than Effective Date. Please re-enter."
7. If Personal Needs Deduction is greater than the Amount, the user receives an error message after pressing the Continue button stating "Personal Need Amount cannot be greater than the Benefit Amount."
8. If the Claim Number has already been used in a previous benefit record, after pressing the Continue button the user receives a notification message stating "Warning! This Claim Number already exists in eWiSACWIS. This Benefit may need to have a unique Claim Number."

9. Drop Down values for: “Benefit is available for (drop down) cost of care expenses”

- Current Month = cd_expns = 2
- Prior Month = cd_expns = 3
- All Outstanding = cd_expns = 1
- No = cd_expns = 0

1.2.3.4. Save Processing

1. There is no save processing on this page.

1.2.3.5. CRUD Matrix

| Table Name | CRUD |
|-------------------|-------------|
| TRUST_ACCOUNT | R |
| BENEFIT | R |

1.2.4. Tab- Ledger

Trust Account - Microsoft Internet Explorer provided by DHFS

eWiSACWIS Print Spell Check Help

Person/Worker Information

Name: Annabel Abby SSN Child: 789-79-7955 Trust Acct #: 9221006 General
Case: Amy Abby SSN Case: 353-41-5655 Closed Date: 00/00/0000 Dedicated
Worker: Frank Fox Phone: Bank: Firststar - Milwaukee - General 12121212

Benefit Record **Ledger**

Account Information

SSI - State SSA - Disability Child Support Other Current Balance: \$400.00
 SSI - Federal SSA - Survivor's VA All Outstanding Cost of Care: \$0.00
 Personal Needs Railroad Retirees Interest

Transaction Ledger Listing

| View | Date | Type | Amount | Detail | ID | Benefit | Description |
|----------------------------------|------------|---------|----------|---------------|----|---------------|-------------|
| <input checked="" type="radio"/> | 08/19/2003 | Deposit | \$150.00 | Adjustment | | Interest | |
| <input type="radio"/> | 08/19/2003 | Deposit | \$250.00 | Deposit Batch | | Child Support | |

Insert

Ledger Record Details

Date: 08/19/2003 Amount: \$150.00 Check ID: Benefit Type: Interest
Type: Deposit Detail: Adjustment Description:

Options: **Go** **Save** **Close**

Done Local intranet

1.2.4.1. Page Overview

The Ledger Tab of the Trust Account page allows the Trust Account Manager to maintain a complete history of all debit and credit transaction for a particular trust account. This ledger can be updated manually, using the Insert button in the Transaction Ledger Listing group box to create a new ledger record. When the Insert button is pressed, the user is navigated to the Maintain Ledger Record page. A ledger record can also be edited or deleted prior to initial save processing by selecting the Edit or Delete hyperlinks, respectively. The Maintain Ledger Record page is discussed in the following section of this document.

The ledger is also updated by the Spread Interest batch, Deposit batch, and the EFT Update batch. The Spread Interest batch will prorate the interest amount specified on the Interest page across all accounts. The Deposit batch will update the ledger by the amount and number of days indicated on the Benefit Record tab. The EFT batch will update the ledger whenever an EFT transaction occurs at the Bank.

To view details of a ledger record, the radio button located to the left of each ledger record is selected. When selected, the associated information appears in the Ledger Record Details group box.

1.2.4.1.1. Withdrawals and Deposits

When a withdrawal is made, it must be made against a specific benefit type. This enables the trust account manager to accurately report how each benefit was spent. In order to determine the remaining balance of a benefit type, the user will select the radio button for that benefit type in the Account Information box. A withdrawal can be made for an amount up to or equal to the amount displayed in the Current Balance field. Withdrawals will be recorded in the table as negative amounts, and deposits will be recorded as positive amounts.

1.2.4.1.2. Maximum Balance

If the current balance of the trust account equals or exceeds the maximum balance level multiplied by the percentage specified on the Maximum Balance page, then a tickler is sent to primary work to initiate a spend down. The primary work must then examine the Trust Account to determine the benefit type and the amount to be spent, and then contact the Trust Account Manager who will process the transaction and write a check. Once the withdrawal is made and the balance is below the maximum specified, then the tickler will be removed.

1.2.4.2. Page Information

Box: Account Information

Fields:

| | |
|-----------------|--|
| SSI-State: | When radio button is selected the current balance will be calculated for the SSI-State benefit and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 2. User selected radio button; Defaults to 'All'. |
| SSI-Federal: | When radio button is selected the current balance will be calculated for the SSI-Federal benefit and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 6. User selected radio button; Defaults to 'All'. |
| Personal Needs: | When radio button is selected the current balance will be calculated for the Personal Needs money and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 3. User selected radio button; Defaults to 'All'. |
| SSA-Disability: | When radio button is selected the current balance will be calculated for the SSA-Disability benefit and this amount will be displayed to the Current Balance field. Calculates |

- balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 1. User selected radio button; Defaults to 'All'.
- SSA-Survivor's: When radio button is selected the current balance will be calculated for the SSA-Survivor's benefit and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 7. User selected radio button; Defaults to 'All'.
- Railroad Retirees: When radio button is selected the current balance will be calculated for the Railroad Retirees benefit and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 8. User selected radio button; Defaults to 'All'.
- Child Support: When radio button is selected the current balance will be calculated for the Child Support benefit and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 5. User selected radio button; Defaults to 'All'.
- VA: When radio button is selected the current balance will be calculated for the VA benefit and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 4. User selected radio button; Defaults to 'All'.
- Interest: When radio button is selected the current balance will be calculated for the Interest money and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 10. User selected radio button; Defaults to 'All'.
- All: When radio button is selected the current balance will be calculated for all money in trust account and this amount will be displayed to the Current Balance field. Calculates balances for all ledger entries for this trust account. User selected radio button; Defaults to 'All'.
- Current Balance: Field displays the current balance based on the radio button selected; System derived; Not user editable; Defaults to current balance of all ledger records.
- Outstanding Cost: Field displays the outstanding cost of

of Care: care for the current participant, case, and county. Sum of PAYMENT.am_nontrust for PAYMENT.id_prsn, PAYMENT.id_case. Only the county's approved non-canceled payments that eligible for trust account withdrawals are included in the sum (Payment.cd_cnty_new= Trust_Account.cd_cnty and Payment.cd_pmnt_stat <> 'C' and Paymnet.id_pmnt in (Select id_wrk_type From Approval where cd_stat='A') and Service_Type.cd_ive_12<>4 (where Service_Type.cd_srvc =Payment.cd_srvc)); System derived; Not user editable.

Box: Transaction Ledger Listing

Fields: View This column has radio buttons available for selection by a worker; When selected, the details from the ledger record are filled into the Ledger Record Details group box; User selected radio buttons; User editable; Defaults to the first record in the repeating group.

Date: Date of the transaction. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.dt_trans.

Type: Displays transaction as either a Deposit or Withdrawal. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.cd_trans_typ.

Amount: The amount of the transaction either negative or positive value. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.am_trans.

Detail: Describes the type of transaction for Deposits (e.g. Monthly Check, One-Time Payment, Adjustment, EFT Deposit, or Interest) and Withdrawal (e.g. Reimbursement, Monthly Check, Manual Check, Adjustment, One-Time Withdrawal or Close Account). System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.cd_detl.

ID: Displays the Check number or other identification number of this payment. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.id_chck_nmbr.

Benefit: Displays the benefit type of this transaction (SSI-State, SSI-Federal, SSA-Disability, SSA-Survivor's, Child Support, VA, Railroad Retirees, Personal Needs, Other Benefits or Interests). System derived; Not user editable.

| | |
|----------------|---|
| | Retrieved from TRUST_ACCOUNT_LEDGER.cd_trans_typ. |
| Description: | Displays description associated with this transaction. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.tx_desc. |
| Box: | Ledger Record Details |
| Fields: | |
| Date: | Date of the transaction. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.dt_trans. |
| Type: | Displays transaction as either a Deposit or Withdrawal. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.cd_trans_typ. |
| Amount: | The amount of the transaction either negative or positive value. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.am_trans. |
| Detail: | Describes the type of transaction for Deposits (e.g. Monthly Check, One-Time Payment, Adjustment, EFT Deposit, or Interest) and Withdrawal (e.g. Reimbursement, Monthly Check, Manual Check, Adjustment, One-Time Withdrawal or Close Account). System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.cd_detl. |
| ID: | Displays the Check number or other identification number of this payment. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.id_chck_nmbr. |
| Benefit: | Displays the benefit type of this transaction (SSI-State, SSI-Federal, SSA-Disability, SSA-Survivor's, Child Support, VA, Railroad Retirees, Personal Needs, Other Benefits or Interests). System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.cd_trans_typ. |
| Reversal: | This check box allows a new ledger record to be automatically inserted for a deposit amount equal to the amount of a batch reimbursement withdrawal. User selected check box; User editable; Defaults to not checked. Maps to TRUST_ACCOUNT_LEDGER.fl_rev. |

| | | |
|-----------------|----------------|--|
| | Description: | Displays description associated with this transaction. System derived; Not user editable. Retrieved from TRUST_ACCOUNT_LEDGER.tx_desc. |
| Links: | Edit: | Opens the Maintain Ledger Record page for the selected ledger record. Only enabled prior to initial save processing. |
| | Delete: | Deletes the selected ledger record from the Trust Account page. Only enabled prior to initial save processing. |
| Options: | Export Ledgers | When selected, the ledger records are exported to a Microsoft Excel document. |
| Buttons: | Insert | Opens the Maintain Ledger Record page for the user to add a new ledger record. Enabled when the Trust Account is open. |
| | Save | Standard Save Processing |
| | Close | Standard Close Processing |

1.2.4.3. Background Processing

1. When the Delete hyperlink is selected for a ledger record, the user is prompted with a notification message that states “Are you sure you want to delete currently selected information?” If the user selects ‘Yes’, the record is removed from the Transaction Ledger Listing group box. The Delete hyperlink is only available prior to initial save processing.
2. If the Trust Account record is closed, the Insert button is disabled. Additionally, the Edit and Delete links are not available for ledger records.
3. When a radio button is selected for a record in the Transaction Ledger Listing group box, the appropriate information is visible in the Ledger Record Details group box.
4. A new ledger record is added by selecting the Insert button in the Transaction Ledger Listing group box. The user is navigated to the Maintain Ledger Record page.
5. The Reversal Button is shown and enabled in the Ledger Record Details group box only:
 - When TRUST_ACCOUNT_LEDGER.cd_trans_type = 2 AND TRUST_ACCOUNT_LEDGER.id_pmnt is <>Null AND TRUST_ACCOUNT_LEDGER.am_trans is => 0 show the Reversal check box with a value of N (not checked).
 - The Reversal Button is not shown

- When TRUST_ACCOUNT_LEDGER.id_pmnt is Null OR TRUST_ACCOUNT_LEDGER.fl_rev = N OR TRUST_ACCOUNT_LEDGER.cd_trans_type = 1 hide the Reversal check box.
6. When the Reversal button is selected (checked)
- Copy TRUST_ACCOUNT_LEDGER row where fl_rev =Y and insert copied row into TRUST_ACCOUNT_LEDGER(row x) and change row x fields TRUST_ACCOUNT_LEDGER.cd_trans_type from a value of 2 to 1, TRUST_ACCOUNT_LEDGER.am_trans from a negative value to a positive value, TRUST_ACCOUNT_LEDGER.fl_rev from a value of Y to N, TRUST_ACCOUNT_LEDGER.cd_detl = 14, and TRUST_ACCOUNT_LEDGER.id_ldgr is a system generated unique id. Also, using the TRUST_ACCOUNT_LEDGER.id_pmnt (where PAYMENT.id_pmnt = TRUST_ACCOUNT_LEDGER.id_pmnt) update the PAYMENT.am_nontrust field with the existing am_nontrust plus TRUST_ACCOUNT_LEDGER.am_trans value on row x AND update PAYMENT.fl_ive_calc = 'Y'.
 - When the Reversal button is unchecked (prior to saving the page)
 - When TRUST_ACCOUNT_LEDGER.fl_rev is changed from a value = Y to a value = N, rollback all changes.

1.2.4.4.Save Processing

1. Save Processing is initiated by clicking on the Save command button. Save processing is also initiated when selecting the Close button, and then answering 'Yes' when the system asks the user if they would like to 'Save changes before closing the page.'
2. When a row is saved, the Edit and Delete hyperlinks are no longer available. No changes can be made to that ledger record.
3. Benefit Type must have an existing benefit on the Benefit Record tab with an effective date else issue error: "Transaction not allowed. You must add an Effective Date to the <benefit type> Benefit record or delete the <benefit type> Ledger record." Interest is the only Benefit Type that this does not apply to.
4. Go through and calculate balance, for this id_trst_acct whose TRUST_ACCOUNT.id_tcklr = null.
 - Using the TRUST_ACCOUNT.id_prsn selected in step 1, sum records TRUST_ACCOUNT_LEDGER.am_trans for TRUST_ACCOUNT_LEDGER.id_prsn and TRUST_ACCOUNT_LEDGER.id_trust_acct = TRUST_ACCOUNT.id_trust_acct and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1, Set this = WS-DEPOSIT-ALL

- Using the TRUST_ACCOUNT.id_prsn select in step 1, sum records TRUST_ACCOUNT_LEDGER.am_trans for TRUST_ACCOUNT_LEDGER.id_prsn and TRUST_ACCOUNT_LEDGER.id_trust_acct = TRUST_ACCOUNT.id_trust_acct and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2, Set this = WS-WITHDRAWAL-ALL
 - Calculate WS-DEPOSIT-ALL minus WS-WITHDRAWAL-ALL = WS-CURRENT-BALANCE
3. If child is FFP (retrieve most recent record ELIG_REDET.cd_stat = 1 and ELIG_REDET.fl_cmplt = Y and ELIG_REDET.fl_void <> Y. (Or if no record on ELIG_REDET, then ELIGIBILITY.cd_stat = 1 and ELIGIBILITY.fl_cmplt = Y and ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and find most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp = "N", and calculate MAX_BALANCE.am_pcntg multiplied by MAX_BALANCE.am_maxbal = WS-FFP-MAX-BAL-NOTIFICATION. If WS-FFP_MAX-BAL-NOTIFICATION =< WS-CURRENT-BALANCE, then if no tickler present, then issue a tickler.
- TICKLER.id_prsn = With TRUST_ACCOUNT.id_case lookup for ASSIGNMENT.id_grp_lvl1 and ASSIGNMENT.cd_asgn_role = '1'. Use this ASSIGNMENT.id_prsn
 - TICKLER.id_tcklr = unique number
 - TICKLER.cd_ctgry = 11
 - TICKLER.cd_type = 2
 - TICKLER.cd_prmry_type = C
 - TICKLER.dt_due = system date
 - TICKLER.id_prmry_type = TRUST_ACCOUNT.id_case
 - TICKLER.tx_dscr = "Trust account for (insert BENEFIT.id_prsn) must be spent down by (insert parmdate + qt_escltn_spv)"
 - TICKLER.dt_escltn_spv = parmdate + CATEGORY_TYPE.qt_escltn_days_spv
 - TICKLER.dt_escltn_pspv = parmdate + CATEGORY_TYPE.qt_escltn_days_psv
 - TICKLER.dt_rmdr = parmdate + CATEGORY_TYPE.qt_lag_days
 - TICKLER.id_cr = id_worker
 - TICKLER.ts_cr = systemdate

- TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr (from above)
4. If child is non-FFP (ELIG_REDET.cd_stat not = 1 and ELIG_REDET.fl_cmplt = Y and ELIG_REDET.fl_void <> Y (or if no record on ELIG_REDET, then ELIGIBILITY.cd_stat not = 1 and ELIGIBILITY.fl_cmplt = Y and ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and find most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp = "Y", and calculate MAX_BALANCE.am_pcntg multiplied by MAX_BALANCE.am_maxbal = WS-FFP-MAX-BAL-NOTIFICATION. If WS-FFP-MAX-BAL-NOTIFICATION =< WS-CURRENT-BALANCE, then if no tickler present, then issue a tickler.
- TICKLER.id_prsn = With TRUST_ACCOUNT.id_case lookup for ASSIGNMENT.id_grp_lv11 and ASSIGNMENT.cd_asgn_role = '1'. Use this ASSIGNMENT.id_prsn
 - TICKLER.id_tcklr = unique number
 - TICKLER.cd_ctgry = 11
 - TICKLER.cd_type = 1
 - TICKLER.cd_prmry_type = C
 - TICKLER.dt_due = system date
 - TICKLER.id_prmry_type = TRUST_ACCOUNT.id_case
 - TICKLER.tx_dscr = "Trust account for (insert BENEFIT.id_prsn) must be spent down by (insert parmdate + qt_escltn_spv)"
 - TICKLER.dt_escltn_spv = parmdate + CATEGORY_TYPE.qt_escltn_days_spv
 - TICKLER.dt_escltn_pspv = parmdate + CATEGORY_TYPE.qt_escltn_days_psv
 - TICKLER.dt_rmdr = parmdate + CATEGORY_TYPE.qt_lag_days
 - TICKLER.id_cr = id_worker
 - TICKLER.ts_cr = systemdate
 - TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr (from above)
5. A tickler check must also be done, and a tickler should be removed if conditions apply. Select a record TRUST_ACCOUNT.id_tcklr not = null. If field is populated, then there is an outstanding tickler.
- Using the BENEFIT.id_prsn selected in step 2, sum records TRUST_ACCOUNT_LEDGER.am_trans for TRUST_ACCOUNT_LEDGER.id_prsn and TRUST_ACCOUNT_LEDGER.id_trust_acct =

TRUST_ACCOUNT.id_trust_acct and

TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1, Set this = WS-DEPOSIT-ALL

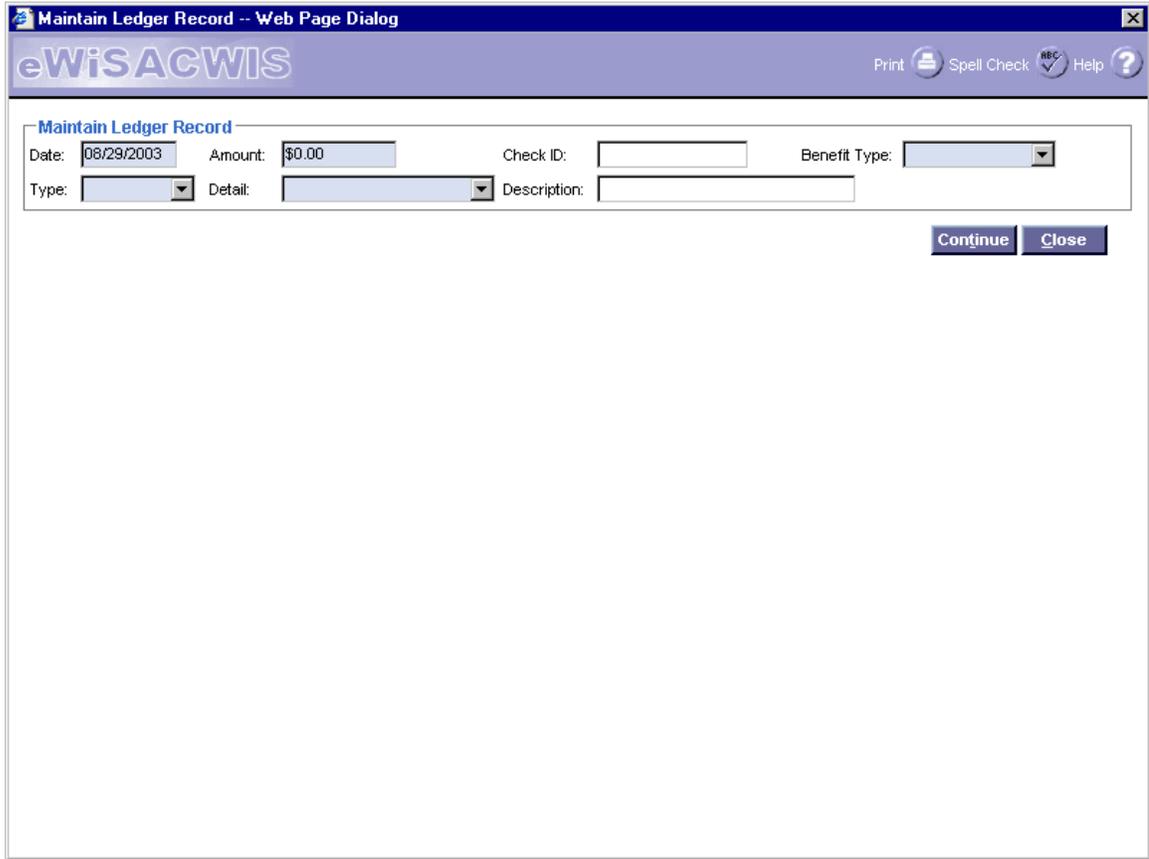
- Using the BENEFIT.id_prsn select in step 1, sum records
TRUST_ACCOUNT_LEDGER.am_trans for
TRUST_ACCOUNT_LEDGER.id_prsn and
TRUST_ACCOUNT_LEDGER.id_trust_acct =
TRUST_ACCOUNT.id_trust_acct and
TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2, Set this = WS-
WITHDRAWAL-ALL
- Calculate WS-DEPOSIT-ALL minus WS-WITHDRAWAL-ALL = WS-
CURRENT-BALANCE
- If Non-FFP (ELIG_REDET.cd_stat not = 1 and ELIG_REDET.fl_cmplt = Y and
ELIG_REDET.fl_void <> Y (or if no record on ELIG_REDET, then
ELIGIBILITY.cd_stat not = 1 and ELIGIBILITY.fl_cmplt = Y and
ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and find
most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp =
“Y”, and calculate MAX_BALANCE.am_pcntg multiplied by
MAX_BALANCE.am_maxbal = WS-FFP-MAX-BAL-NOTIFICATION. If WS-
FFP-MAX-BAL-NOTIFICATION > WS-CURRENT-BALANCE, delete record
for TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr, and move spaces to
TRUST_ACCOUNT.id_tcklr
- If FFP (retrieve most recent record ELIG_REDET.cd_stat = 1 and
ELIG_REDET.fl_cmplt = Y and ELIG_REDET.fl_void <> Y. (Or if no record
on ELIG_REDET, then ELIGIBILITY.cd_stat = 1 and ELIGIBILITY.fl_cmplt =
Y and ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and
find most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp =
“N”, and calculate MAX_BALANCE.am_pcntg multiplied by
MAX_BALANCE.am_maxbal = WS-FFP-MAX-BAL-NOTIFICATION. If WS-
FFP_MAX-BAL-NOTIFICATION > WS-CURRENT-BALANCE, delete record
for TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr, and move spaces to
TRUST_ACCOUNT.id_tcklr

1.2.4.5.CRUD Matrix

| Table Name | CRUD |
|----------------------|------|
| TRUST_ACCOUNT_LEDGER | CRU |
| TRUST_ACCOUNT | CRU |
| MAX_BALANCE | R |

| | |
|-------------|-----|
| TICKLER | CRD |
| ELIGIBILITY | R |
| ELIG_REDET | R |

1.2.5. Page- Maintain Ledger Record



1.2.5.1. Page Overview

Navigation

This page is accessed by selecting the Insert button on the Ledger tab or by selecting an Edit hyperlink to the right of a ledger record.

Page Summary

The Maintain Ledger Record page is used to enter new ledger records to the Ledger tab. Additionally, a user can update ledger records on this page. When the user selects the Continue button, the Maintain Ledger Record page will close and the new or updated information will be passed back to the Ledger tab. This information will be saved to the database when the Save button is selected on the Trust Account page. However, business logic errors will be validated when the user selects the Continue button.

1.2.5.2. Page Information

Box: Maintain Ledger Record

| | | |
|-----------------|---------------|---|
| Fields: | Date: | Date of the transaction. User entered date field; User editable; Required; Defaults to current date. Maps to TRUST_ACCOUNT_LEDGER.dt_trans. |
| | Type: | User must select transaction as either a Deposit or Withdrawal. User selected drop down box; User editable; Required; No default value. Maps to TRUST_ACCOUNT_LEDGER.cd_trans_typ. |
| | Amount: | The amount of the transaction. This amount is either negative value for withdrawals or a positive value for deposits. User entered amount field; User editable; Required; No default value. Maps to TRUST_ACCOUNT_LEDGER.am_trans. Required. |
| | Detail: | Describes the type of transaction for Deposits (e.g. Monthly Check, One-Time Payment, Adjustment, EFT Deposit, or Interest) and Withdrawal (e.g. Reimbursement, Monthly Check, Manual Check, Adjustment, One-Time Withdrawal or Close Account). User selected drop down box; User editable; Required; No default value. Maps to TRUST_ACCOUNT_LEDGER.cd_detl. Required. |
| | ID: | Displays the Check number or other identification number of this payment. User entered text field; User editable; Not required; No default value. Maps to TRUST_ACCOUNT_LEDGER.id_chck_nmbr. |
| | Benefit Type: | Displays the benefit type of this transaction (SSI-State, SSI-Federal, SSA-Disability, SSA-Survivor's, Child Support, VA, Railroad Retirees, Personal Needs, Other Benefits or Interests). User selected drop down box; User editable; Required; No default value. Maps to TRUST_ACCOUNT_LEDGER.cd_trans_typ. Required. |
| | Description: | Displays description associated with this transaction. User entered text field; User editable; Not required; No default value. Maps to TRUST_ACCOUNT_LEDGER.tx_desc. |
| Buttons: | Continue | Runs business logic checks for the ledger record. If there are no business errors, the user is navigated back to the Trust Account page. The newly entered information is passed back to the Trust Account page, Ledger tab. |
| | Close | Standard Close Processing |

1.2.5.3. Background Processing

1. When Type = Deposit, then Amount must be positive. If the amount is negative and the user selects the Continue button, the worker will receive an error message stating “Must enter a Positive Value or Transaction must be a Withdrawal.”
2. When Type = Withdrawal, then Amount must be negative. If the amount is positive and the user selects the Continue button, the worker will receive an error message stating “Must enter a Negative Value or Transaction must be a Deposit.”
3. When Type = Deposit, filter is applied to Detail and only display: Adjustment, Check Reversal, Deposit Batch, Prior Period Reversal, Transfer from Dedicated, Monthly Check, One Time Payment, Overpayment Adjustment, EFT Deposit, and Interest.
4. When Type = Withdrawal, filter is applied to Detail and only display: Reimbursement, Adjustment, Check Reversal, Close Account, Prior Period Reversal, Transfer from Dedicated, Manual Check, Monthly Check, One Time Withdrawal, Overpayment Adjustment, and EFT Withdrawal.
5. If Benefit Type is changed, and Amount is set back to zero.
6. The Type, Amount, Detail, and Benefit Type fields are required and cannot have a null value at time of selecting the Continue button. If any of these fields are null, the user will receive an error validation stating “Please enter data in the fields which are highlighted - <field name>.”

1.2.5.4. Save Processing

1. There is no save processing on this page.

1.2.5.5. CRUD Matrix

| Table Name | CRUD |
|----------------------|------|
| TRUST_ACCOUNT_LEDGER | R |
| TRUST_ACCOUNT | R |

1.2.6. Page - Bank

Select Bank - Microsoft Internet Explorer provided by DHFS

eWiSACWIS Print Spell Check Help

Bank Selection

Select Bank:

Milwaukee - General

Continue Close

Done Local intranet

Bank - Microsoft Internet Explorer provided by DHFS

eWiSACWIS Print Spell Check Help

Bank Account Information

County: Milwaukee

Bank Name: Firststar

ABA Number: 12345 Account Number: 12121212

General Dedicated

Bank Address

C/O: Joe Smith

Street: 111 State Street Suite: 222

City: Madison State: WI Zip: 53701 Country:

Effective Date: 01/01/2000

Options: Go

Save Close

Done Local intranet

1.2.6.1. Page Overview

Navigation

The Bank page is accessed through the Maintain > Bank from the main application menu bar. After accessing this menu option, the Select Bank intermediary page opens. The user must select the county and bank type (general or dedicated) from the Select Bank drop down box. After selecting a value and pressing the Continue button, the user is navigated to the Bank page. From this page the user can access the Interest pop-up page where user can maintain a history of interest earned at this bank. (See Section 1.2.7 for details on the Interest page.)

Page Summary

The Bank page allows the Trust Account Manager to maintain current address and account information for each of the county's bank where trust account money is kept. Each county has the ability to create two separate bank accounts for trust funds. The General Account will hold all benefits, and will be accessed by the Trust Account Draw Down batch for State reimbursement (See *FM04a: Reimbursement*). The Dedicated Account will hold SSI benefits that have been paid in a sum that is greater than six months. The Dedicated Account will not be accessed by the Trust Account Draw Down batch. The Trust Account manager will manually transfer money from the Dedicated Account into the General Account when the money has been approved for State reimbursement. Each child's Trust Account must be matched to one account number. This matching occurs when the Trust Account page is opened for the first time when creating a child's trust account.

1.2.6.2. Page Information

Box: Bank Selection (on Select Bank page)

Fields: Select Bank: This drop down lists all Wisconsin counties along with both general and dedicated bank account types. Concatenated and retrieved from id_grp = REGION and on the CODE_DESC table.

Box: Bank Account Information (on Bank page)

Fields: County: The name of the county associated with the bank account record; System derived; Not user editable.

Bank Name: Name of the bank for which this account is created. User entered text field; User editable; Required; No default value. Retrieved from the tx_name on the BANK table.

ABA Number: ABA routing number for this bank. User entered text field; User editable; Required; No default value. Retrieved from cd_aba on the BANK table.

Account Number: Account number for this trust account. User entered text field; User editable; Required; No default value. Retrieved from cd_acct_number on the BANK table.

General: Radio button indicates this address and account is for the dedicated Trust Account fund; System derived; Not user editable. Retrieved from the BANK table where cd_type = 1.

Dedicated: Radio button indicates this address and account is for the dedicated Trust Account fund; System derived; Not user editable. Retrieved from the BANK table where cd_type = 2.

Box: Bank Address (on Bank page)

Fields: C/O: Care-Of Name for this bank address. User entered text field; User editable; Not required; No default value. Retrieved from the ad_ln2 on the ADDRESS table where ADDRESS.cd_grp = 12 and cd_grp_type = 1.

Street: Street number and street name for this bank. User entered text field; User editable; Not required; No default value. Retrieved from the ad_strt_nbr and ad_strt_nme on the ADDRESS table where ADDRESS.cd_grp = 12 and cd_grp_type = 1.

Suite: Suite number for this address. User entered text field; User editable; Not required; No default value. Retrieved from the ad_aprt on the ADDRESS table where ADDRESS.cd_grp = 12 and cd_grp_type = 1.

WI City: Contains the concatenation of the city name and zip code for Wisconsin cities. When selected, the City, State, and Zip fields fill with the selected information. User selected drop down box; User editable; Not required; No default value.

City: City for this address. User entered text field; User editable; Required; No default value. Retrieved from the tx_town on the ADDRESS table where ADDRESS.cd_grp = 12 and cd_grp_type = 1.

State: State for this address. User selected drop down box; User editable; Not required; No default value. Retrieved from the cd_state on the ADDRESS table where ADDRESS.cd_grp = 12 and cd_grp_type = 1.

Zip: Zip number for this address. User entered numeric field; User editable; Not required; No default value. Retrieved from the cd_zip on the ADDRESS table where

| | |
|-----------------|--|
| | ADDRESS.cd_grp = 12 and cd_grp_type = 1. |
| Country: | Country for this address. User selected drop down box; User editable; Not required; Defaults to United States. Retrieved from the ADDRESS table where ADDRESS.cd_grp = 12 and cd_grp_type = 1. |
| Effective Date: | Date on which this address become effective. User entered date field; User editable; Required; Defaults to 00/00/0000. Retrieved from the dt_efct_start on the ADDRESS table. |
| Options: | |
| Interest: | Navigates the user to the Interest page for the given bank account on the Bank page. |
| Check Update: | Navigates the user to the Check Update page for the given bank account on the Bank page. |
| Buttons: | |
| Continue | On the Select Bank page. Navigates the user to the Bank page for the selected bank account. |
| Close | Standard Close Processing on the Select Bank page |

1.2.6.3. Background Processing

1. After a county and account type is selected on the Select Bank page and the Continue button is pressed, the address and account information for this id_bank is retrieved.
 - a. If a general bank account is selected, the address and account information is brought forward for BANK.cd_type = 1.
 - b. If a dedicated bank account is selected, the address and account information is brought forward for BANK.cd_type = 2.

1.2.6.4. Save Processing

1. Save Processing is initiated by clicking on the Save command button on the Bank page. Save processing is also initiated when selecting the Close button, and then answering 'Yes' when the system asks the user if they would like to 'Save changes before closing the page.'
2. A unique id_bank is created and saved to BANK table when a new county is selected from the drop down for each account type (General or Dedicated).
 - a. If the Save button is pressed for a general bank account on the Bank page, the record is saved to the BANK table where BANK.cd_type = 1.
 - b. If the Save button is pressed for a dedicated bank account on the Bank page, the record is saved to the BANK table where BANK.cd_type = 2.

3. The Bank Name, ABA Number, Account Number, City, and Effective Date fields are required and cannot have a null value at time of selecting the Save button on the Bank page. If any of these fields are null, the user will receive an error validation stating “Please enter data in the fields which are highlighted - <field name>.”
4. If the Interest value is selected from the Options drop down box for a Bank record that has not been initially saved, the user will receive a notification message stating “The current Bank record must be saved before proceeding. Do you wish to save the Bank record?”
5. If the Check Update value is selected from the Options drop down box for a Bank record that has not been initially saved, the user will receive a notification message stating “The current Bank record must be saved before proceeding. Do you wish to save the Bank record?”

1.2.6.5.CRUD Matrix

| Table Name | CRUD |
|-------------------|-------------|
| BANK | CRU |
| ADDRESS | CRU |
| CODE_DESC | R |

1.2.7. Pop-up - Interest

Interest -- Web Page Dialog

eWiSACWIS Print Spell Check Help

Bank

County: Milwaukee Total Balance: \$12,335.00

General Dedicated

Interest Details

| Effective Date | Interest Amount | Flag Posted | Remainder |
|----------------|-----------------|--------------------------|-------------------------------|
| 01/01/2003 | \$200.00 | <input type="checkbox"/> | \$0.00 Delete |

[Insert](#)

[Recalculate](#) [Save](#) [Close](#)

1.2.7.1. Page Overview

The Interest page allows the trust account manager to post accrued interest to the trust accounts for each child. The Interest page is accessed from the Bank page by selecting the Interest value from the Options drop down box and pressing the Go button. When the bank statement is received by the Trust Account Manager, the amount of interest earned can be entered into the Interest page along with the effective date. The effective date should be the date on which the statement was produced. When the Spread Interest batch process calculates each child's portion of the interest, it will calculate the balance based on ledger transactions that are less than or equal to the effective date. It is therefore crucial that the transaction date and interest effective date are accurately entered into the system. When the page is closed the effective date is frozen. After the spread interest batch runs, the Flag Posted field will be checked by the batch and the Interest Amount will be frozen.

1.2.7.1.1. Correcting Wrong Interest Amounts

If after the spread interest batch has run, the trust account manager discovers the incorrect interest amount, the amount can be re-entered and re-spread across accounts. By un-checking the Flag Posted check box, the amount field is enabled and the user can re-enter the interest amount for that effective date. The following batch cycle will

pick up this new amount and recalibrate each child's portion of interest. If the effective date was entered incorrectly, the amount of interest for that date must be set to zero, and a new row needs to be inserted into the Interest page.

1.2.7.1.2. Current Account Balance Calculation

From this page a reconciliation against the current balance at the bank can be performed. The Total Balance field contains the current balance of all ledger transactions for this bank account. By summing the interest rows where Flag Posted is not checked, and adding this amount to Total Balance, one is able to determine the current balance and confirm that it matches the balance as determined by the bank.

1.2.7.2. Page Information

Box: Interest

Fields: County: Displays county selected on the Bank page. System derived; Not user editable. Retrieved from id_grp = REGION on the CODE_DESC table.

Total Balance: Displays current balance for this account number. System derived; Not user editable. Calculated based on TRUST_ACCOUNT_LEDGER.am_trans for this id_bank.

General/Dedicated: Displays the bank account type from the Bank page; System derived; Not user editable. Retrieved from BANK.cd_type.

Box: Interest Details

Fields: Effective Date: Date used for the calculation of the total balance and each child's balance, in order to determine the prorated interest amount for each child. User entered date field; User editable until initial save processing; Required; Defaults to 00/00/0000. Retrieved from dt_effctv on the TRUST_ACCOUNT_LEDGER table.

Interest Amount: Amount of interest accrued for this effective date. User entered numeric field; User editable unless the Flag Posted checkbox is checked via the Spread Interest batch; Required; No default value. Retrieved from am_intrst on the TRUST_ACCOUNT_INTEREST table.

Flag Posted: Indicates that this interest amount has been spread across the individual trust accounts by the spread interest batch process. The user is able to uncheck box, but not place a check in the box. The box is only checked by the batch process. User selected check box; User editable once it is

| | | |
|-----------------|--------------|---|
| | | checked via the Spread Interest batch; Defaults to not checked and disabled. Retrieved from fl_pstd on the TRUST_ACCOUNT_INTEREST table. |
| | Remainder: | Stores the portion of the interest amount that could not be divided among trust account records via the Spread Interest batch; System derived; Not user editable; No default value. |
| Links: | Delete: | Deletes the selected interest record from the Interest Details group box. |
| Buttons: | Insert | Adds a new interest record to the Interest Details group box. |
| | Recalculate: | Performs a recalculation of account Total Balance for records where Flag Posted is not checked. |
| | Save | Standard Save Processing |
| | Close | Standard Close Processing |

1.2.7.3. Background Processing

1. When the Insert button is selected, a new row is added to the Interest Details group box. The Effective Date and Interest Amount fields are enabled and required.
2. When the Delete hyperlink is selected for an interest record, the user is prompted with a notification message that states “Are you sure you want to delete currently selected information?” If the user selects ‘Yes’, the record is removed from the Interest Details group box.
3. If Flag Posted is checked, Interest Amount is disabled.

1.2.7.4. Save Processing

1. Save Processing is initiated by clicking on the Save command button. Save processing is also initiated when selecting the Close button, and then answering ‘Yes’ when the system asks the user if they would like to ‘Save changes before closing the page.’
2. When the Recalculation button is selected, the user is prompted to save the Interest page. If the user opts the save the page, the interest records are written to the TRUST_ACCOUNT_INTEREST table and the Total Balance field is recalculated to temporarily include the un-posted interest rows.
3. When row is saved, Effective Date becomes disabled.
4. If two records have the same Effective Dates, on save processing the user will receive an error message stating “Interest for the same day cannot be posted.”

5. When saving dt_effctv, save current time next to date.
6. The Effective Date and Interest Amount fields are required and cannot have a null value at time of selecting the Save button. If either of these fields are null, the user will receive an error validation stating “Please enter data in the fields which are highlighted - <field name>.”

1.2.7.5.CRUD Matrix

| Table Name | CRUD |
|------------------------|-------------|
| TRUST_ACCOUNT_INTEREST | CRU |
| TRUST_ACCOUNT_LEDGER | R |
| BANK | R |
| TRUST_ACCOUNT | R |
| CODE_DESC | R |

1.2.8. Page – Maximum Balance

| | Effective Date | Maximum Balance | Percentage Amount% | |
|---------|---|-------------------------------------|-----------------------------------|------------------------|
| FFP | 06/05/2002 | \$1,500.00 | 100.00% | |
| Non-FFP | 06/05/2002 | \$1,500.00 | 100.00% | |
| FFP | 01/01/2006 | \$2,000.00 | 100.00% | |
| Non-FFP | 01/01/2006 | \$2,000.00 | 100.00% | |
| Non-FFP | <input type="text" value="00/00/0000"/> | <input type="text" value="\$0.00"/> | <input type="text" value="0.00"/> | |
| FFP | <input type="text" value="00/00/0000"/> | <input type="text" value="\$0.00"/> | <input type="text" value="0.00"/> | Delete |

Insert

Save Close

1.2.8.1. Page Overview

Navigation

The Maximum Balance page is opened by selecting the Maintain > Maximum Balance value from the main application menu bar.

Page Summary

This page allows Trust Account Manager to maintain maximum balance amount for children with trust accounts. Children with trust accounts are distinguished as either one of two types. Federal Financial Participation (FFP) children are those who are Eligible and Reimbursable. Non-FFP children are those who are 1) Eligible and Non-Reimbursable or 2) Ineligible. A history of maximum balance amounts are stored on the Maximum Balance page. Selecting Insert button from the Maximum Balance group box creates two new rows. The first row is the maximum balance for non-FFP children and the second row is for FFP children. The Trust Account Manager must fill in an effective date from which this maximum balance will be applied. The user must also enter the maximum balance amount and the percentage at which a notification will be sent to primary worker. For example, if the maximum balance is

set to \$2000 and percentage is 75%, then when a child's trust account balance is equal to or greater than \$1500, a tickler will be sent to worker.

1.2.8.2. Page Information

Box: Maximum Balance

Fields: Effective Date: Date from which the maximum balance amount is effective. User entered date field; User editable before save process; Required; Defaults to 00/00/0000. Retrieved from dt_effctv on the MAX_BALANCE table.

Amount: The maximum balance amount for this FFP or Non-FFP category. User entered numeric field; User editable before save process; Required; No default value. Retrieved from the PERSON table using id_cr on the PAYMENT record.

Percentage: The percentage of the maximum balance at which a tickler will be issued. User entered numeric field; User editable before save process; Not required; No default value. Retrieved from dt_rqst on the PAYMENT table.

Links: Delete: Deletes the maximum balance record. Available prior to initial save processing.

Buttons: Insert Inserts a new non-FFP row and an FFP row in the Maximum Balance group box.

Save Standard Save Processing

Close Standard Close Processing

1.2.8.3. Background Processing

1. When the Delete hyperlink is selected for a maximum balance record, the user is prompted with a notification message that states "Are you sure you want to delete currently selected information?" If the user selects 'Yes', the record is removed from the Maximum Balance group box. The Delete hyperlink is only available prior to initial save processing.
2. When the Insert button is selected, a new row is added to the Maximum Balance group box. The Effective Date and Maximum Balance fields are enabled and required. The Percentage Amount field is enabled.

1.2.8.4. Save Processing

1. Save Processing is initiated by clicking on the Save command button. Save processing is also initiated when selecting the Close button, and then answering 'Yes' when the system asks the user if they would like to 'Save changes before closing the page.'
2. The Effective Date and Maximum Balance fields are required and cannot have a null value at time of selecting the Save button. If either of these fields are null, the user will receive an error validation stating "Please enter data in the fields which are highlighted - <field name>."

1.2.8.5. CRUD Matrix

| Table Name | CRUD |
|-------------------|-------------|
| MAX_BALANCE | CRU |

1.2.9. Page – Closed Cases Trust Account Search Page

The screenshot shows a web browser window titled "Closed Cases Trust Account Search - Microsoft Internet Explorer provided by DHFS". The page header includes the eWiSACWIS logo and navigation links for Print, Spell Check, and Help. The main content area is divided into two sections: "Search Criteria" and "Results".

Search Criteria:

Last Name: First Name: Sounds Like:

Person ID: County:

Results:

| Doe, Baby | | Person ID: 9221153 | | | |
|-----------|-----------|--------------------|-----------|-----------------------------|--|
| Case | Doe, Baby | 9221032 | Milwaukee | Trust Acct. | |
| Doe, John | | Person ID: 9221155 | | | |
| Case | Doe, John | 9221034 | Milwaukee | Trust Acct. | |

The browser status bar at the bottom shows "Done" and "Local intranet".

1.2.9.1. Page Overview

Navigation

This page is accessed through the main menu by selecting Maintain > Financial Download > Trust Accounts.

Page Summary

The Closed Cases Trust Account search page allows workers to search for general trust accounts in the database for closed cases. This page also allows workers to create a general trust account if one does not exist. Workers will not be able to create a new general trust account if one already exists for that case, child, and county. eWiSACWIS displays the results in rows of data. The worker enters basic criteria contained within the search criteria group box and clicks the Search button to perform the search. The database is searched for data that meets the search criteria entered and the results of the search are returned and displayed in the Results group box. The results will be returned in two parts: first, all cases not associated with a general trust account and second, existing general trust accounts.

1.2.9.2. Page Information

| | | |
|-----------------|-----------------|--|
| Box: | Search Criteria | |
| Field: | Last Name: | Last name of the search participant; partial searches supported using the * wildcard character; partial searches not supported with sounds like search; can be used as the sole search criterion; User entered text field; User editable; Required; No default value. |
| | First Name: | First name of the search participant; partial searches supported using the * wildcard character; partial searches not supported with sounds like search; User entered text field; User editable; Not required; No default value. |
| | Sounds Like: | The phonetic equivalent of the person first name and the person last name; check box; checkbox defaults to checked; if the sounds like check box is checked, partial name searches will not be supported; User selected check box; User editable; Not required; Defaults to checked. |
| | Person ID: | Specifies a unique person ID (eWISACWIS generated person ID); can be used as the sole search criterion; User entered numeric field; User editable; Not required; No default value. |
| | County: | This drop down field pre-fills based on the county of the worker (cd_ofc_div) who has accessed the page. User selected drop down box; User editable; Not required; Defaults to the county of the logged in worker. |
| Links: | Trust Acct. | Opens a trust account record for a person returned during the closed case search. |
| Buttons: | Search | This button will initiate a search of the database based on the criteria entered in the search criteria structure. Disabled until data is entered in the search criteria. |
| | Close | Closes the page and returns the user to the Desktop. |

1.2.9.3. Background Processing

1. One of the following elements is necessary on all searches: last name or person ID. If data is entered in the Person ID field, all other criteria fields are disabled and blanked out. If data is entered in this field and then deleted, all other fields are enabled.
2. Partial searches on names will be supported in the form ABC* where ABC corresponds to the first 3 letters of the name and the * is a wildcard character. A minimum of 1 character and one wildcard character may be used. The wildcard character can be used in both name fields (Last Name and First Name) in the same search. The system will not perform partial

name searches when Sounds Like is selected.

3. A search is initiated for **last name** by clicking the Search button. Based on the number of records that are returned the outliner will display the data as follows:
 - a. Display the results on the Closed Cases Trust Account outliner if CASE.cd_case_stat = 'C'.
 - b. If the last name is not associated with a trust account then a case data row will be displayed under the person name: PERSON.nm_lst || ' ' || PERSON.nm_frst, PERSON.id_prsn. Display: 'Case', CASE.nm_case_lst || ' ' || CASE.nm_case_frst, CASE.id_case, Trust_Account.cd_cnty.
 - c. If the last name is associated with a trust account then a data row will be displayed under the person name: PERSON.nm_lst || ' ' || PERSON.nm_frst, PERSON.id_prsn. Display: 'Trust Account', 'Open' (TRUST_ACCOUNT.dt.closing <> Null) or 'Closed' (TRUST_ACCOUNT.dt.closing = Null), CASE.id_case, Trust_Account.cd_cnty.
4. The county field pre-fills to the county of the worker accessing the page. This field is modifiable and can be used or not used to limit the search results.
5. A search is initiated for **Person ID** by clicking the Search button. Based on the number of records that are returned the outliner will display the data as follows:
 - a. Display the results on the Closed Cases Trust Account outliner if CASE.cd_case_stat = 'C'.
 - b. If the id_prsn is not associated with a trust account then a case data row will be displayed under the person name: PERSON.nm_lst || ' ' || PERSON.nm_frst, PERSON.id_prsn. Display: 'Case', CASE.nm_case_lst || ' ' || CASE.nm_case_frst, CASE.id_case, Trust_Account.cd_cnty.
 - c. If the id_prsn is associated with a trust account then a data row will be displayed under the person name: PERSON.nm_lst || ' ' || PERSON.nm_frst, PERSON.id_prsn. Display: 'Trust Account', 'Open' (TRUST_ACCOUNT.dt.closing <> Null) or 'Closed' (TRUST_ACCOUNT.dt.closing = Null), CASE.id_case, Trust_Account.cd_cnty.
6. All entered criteria are used to determine the appropriate result set.
7. If a search fails, a message box stating "No closed case found for the criteria specified." will be displayed.
8. The Trust Account hyperlink associated to a person for a case in the search results can either open an existing general Trust Account page or create a new general Trust Account page.

1.2.9.4.Save Processing

None

1.2.9.5. CRUD Matrix

| Table Name | CRUD |
|-------------------|-------------|
| CASE | R |
| PERSON | R |
| TRUST_ACCOUNT | CR |
| WORKER | R |

1.2.10. Page – Non-WiSACWIS Benefits Page

1.2.10.1. Page Overview

This page is access from the Maintain menu on the eWiSACWIS desktop. The Non-WiSACWIS Benefits page allows workers to search and view records where trust funds were received for Non-WiSACWIS children. This information could then be reported to BFS. The only money reported on this page is for children who are known to be FFP. HSRS/BFS staff could select a particular county or all counties based on what is selected in the View By field and/or select records entered between particular dates. The information retrieved can be downloaded to a spreadsheet using the selection from the Options dropdown.

1.2.10.2. Page Information

| Box | Search Criteria |
|---------------|---|
| Field: | View By: This field can display all the data entered or a particular county's data; worker selected drop down field. Enabled. |
| | View Records From: This date field will display county data entered after the specified date; worker entered; mandatory; enabled. |
| | View Records To: This date field will display county data entered until the specified date; worker entered; enabled. |

| Box | Non-WiSACWIS Benefits Results |
|---------------|---|
| Date Created: | The date on which the record is created. System derived; Not user editable. Retrieved from NON_SACWIS_BENEFITS.dt_appl. |
| Last Name: | Last name of the non-WiSACWIS child; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.nm_lst. |
| First Name: | First name of the non-WiSACWIS child; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.nm_frst. |
| SSN | Social security number of the non-WiSACWIS child; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.id_ssn. |
| County | County of the non-WiSACWIS child; pre-fills based on the county of the worker entering the data. System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.cd_cnty. |

| | |
|------------------|--|
| FFP Status | FFP status (Eligible & Reimbursable, Eligible & Not-Reimbursable, Ineligible, Unknown) of the non-WiSACWIS child; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.cd_stat. . |
| Refund Amount | Amount of the benefit being recorded; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.amt_rcvd. |
| Facility Type | Facility type (Group Home, CCI, FH, Adoption, Kinship, and Unknown) associated with the non-WiSACWIS child; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.cd_fcly_type. |
| Date App. To: | Date the benefit is applicable towards; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.dt_appl. |
| Service Category | Service Category of the service provided by the provider; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.cd_srvc_ctgry. |
| Service Type | Service Type of the service provided by the provider; System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.cd_srvc. |
| Facility Name | Name of the provider providing out of home care. System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.tx_prvd_org. |
| Provider ID | WiSACWIS ID of provider identified as providing services for the child. System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.id_prvd_org. |
| Maint. Amt. | Maintenance Amount that can be claimed. Used by BFS. System-derived. Not User-editable. Retrieved from NON_SACWIS_BENEFITS.am_maint. |
| Title XIX Amt. | Title XIX Amount that can be claimed. Used by BFS. System-derived. Not user-editable. Retrieved from NON_SACWIS_BENEFITS.am_titlexix. |

| | | |
|-----------------|--------|---|
| Buttons: | Search | Standard Search Processing based on criteria specified in Search Criteria groupbox. |
| | Close | Standard Close Processing |
| | Insert | When selected, the Maintain NonWiSACWIS Benefit Record page pops up. |

| | | |
|----------------|-----------------------------------|---|
| Options | Export Non-WiSACWIS Benefits List | Saves the Non-WiSACWIS Benefits records in a spreadsheet. |
|----------------|-----------------------------------|---|

1.2.10.3. Background Processing

- When the page is opened the view as field will default to a null value and no data rows will be displayed.
- When the view by field is selected, all rows associated with the county selection will be displayed. If the ‘All Counties’ selection is used, all data will be displayed.
- The View Records To field cannot be greater than the current date.
- The View Records From date cannot be greater than the current date.
- The View Records From Date cannot be after the View Recordes To Date.
- If no View Records To date is specified, the end date will be the current date.
- Records will be retrieved based on the Date Creation being within the specified From and To dates.
- Records are retrieved after the Search button is selected.
- If the user attempts to search information without saving the results from the prior search call, provide the standard unsaved change alert (“Do you wish to save your changes”), allowing the user to save, not save or cancel their search action.
-
- When the worker selects the Insert button, the Maintain Non-WiSACWIS Benefits page pops up.
- When the worker selects the Export Non-WiSACWIS Benefits List from the Options dropdown, the data retrieved is saved in the C:/ewisacwis folder. The data is saved in a downloadable format. The column names in the spreadsheet match the column names displayed online.
-
-
-

1.2.10.4. Save Processing

- Standard Save processing.

1.2.10.5. CRUD Matrix

| Table Name | CRUD |
|---------------------|------|
| NON_SACWIS_WISACWIS | CRUD |

1.2.11. Page – Maintain Non-WiSACWIS Benefits Page

1.2.11.1. Page Overview

The Maintain Non-WiSACWIS Benefits page allows workers to record and delete information on Non-WiSACWIS children who receive trust funds. This information could then be reported to BFS. The only money recorded on this page is for children who are known to be FFP. Once the information is entered, the system will calculate the Maintenance and Title XIX amounts for the eligible children. Once saved, the information entered in this page will be displayed in the Results group box of the Non-WiSACWIS Benefits page.

1.2.11.2. Page Information

| Box | Non-WiSACWIS Benefits | |
|---------------|------------------------------|--|
| Field: | Date Created | This field can display all the data entered or a particular county's data; worker selected drop down field. Enabled. |
| | Last Name | This date field will display county data entered after the specified date; worker entered; mandatory; enabled. |
| | First Name | This date field will display county data entered until the specified date; worker entered; enabled and mandatory. |
| | SSN | Social Security number of the child; user-entered; enabled and modifiable. Mandatory. |
| | County | County of the non-WiSACWIS child; pre-fills with the county of the worker creating the record. Enabled and modifiable. Mandatory. |
| | FFP Status | FFP status (Eligible & Reimbursable, Eligible & Not-Reimbursable, Ineligible, Unknown) of the non-WiSACWIS child; user-entered; enabled and modifiable; mandatory. |
| | Service Category | Service category of the placement for the non-WiSACWIS child applicable to the trust funds received. Selected through the search functionality. System-derived. Not editable on the Maintain Non-WiSACWIS Benefits page. |
| | Service Type | Service type of the placement for the non-WiSACWIS child applicable to the trust funds received. Selected through the search functionality. System-derived. Not editable on the Maintain Non-WiSACWIS Benefits page. |
| | Facility Name | WiSACWIS provider selected through the search functionality. System-derived. Not user-editable. Pre- |

| | | |
|-------------------|-----------|---|
| | | filled from PROVIDER.tx_prvd_nm |
| Provider ID | | WiSACWIS ID of the provider selected through the search criteria; system-derived. Not editable. Pre-filled from PROVIDER.id_prvd_org. |
| Facility Type | | Facility type (Group Home, CCI, FH, Adoption, Kinship, and Unknown) associated with the non-WiSACWIS child; user-entered dropdown; enabled and modifiable; mandatory. |
| Date App. To | | Date the benefit is applicable towards; user-entered; modifiable and enabled. Mandatory. |
| WI Facility Name | | Non-WiSACWIS facility in which the child had been placed. User-entered. Modifiable and enabled. Mandatory. |
| WI Service Type | | Non-WiSACWIS service type associated with the placement and the facility in which the child resided. User-entered. Modifiable and enabled. Mandatory. |
| Refund Amt. | | Refund amount received that is applicable to the non-WiSACWIS child's trust account. User-entered. Modifiable and enabled. Mandatory. |
| Maint. Amt. | | Maintenance amount that can be used for claiming purposes for children placed in foster homes, group homes and RCCs. Not user-editable. Pre-filled from NON_SACWIS_BENEFITS.am_maint. Amount calculated when the Calculate button is clicked. |
| Title XIX Amt. | | Title XIX amount that can be used for claiming purposes for children placed in RCCs. Not user-editable. Pre-filled from NON_SACWIS_BENEFITS.am_titlexix. Amount calculated when the Calculate button is clicked. |
| Hyperlink: | Search | Opens the Service Category/Service Type screen for the user to search for providers associated with Service Category and Service Type specified. |
| Buttons: | Calculate | Enabled when refund amount is entered and the provider selected is a WiSACWIS provider. Calculations are based on criteria specified in Background Processing. |
| | Save | Standard Save Processing. |
| | Close | Standard Close Processing |

Background Processing:

- The page opens up when the user selects the Insert button on the main page.
- If Facility Name and Service Type are user-entered, the Maint. Amt and Title XIX Amt fields are disabled and no amount is calculated and pre-filled.
- The Save button saves the information and populates the results in the Results group box of the main page.
- The Close button performs the standard Save and Close procedures, prompting the users to save the information entered. The recently entered information is displayed in the main page.

- The Provider Search hyperlink will allow the user to specify the Service Category and Service Type using the standard Provider Search criteria.
 - The user selects the Service Category based on a drop down menu. The options provided to the user are filtered based on the county of the worker.
 - The user selects the Service Type based on a drop down menu. The values for the Service Type drop down are filtered based on the Service Category and County selected.
 - The provider selected by the user pre-fills into the Facility Name field.
 - The corresponding Provider ID pre-fills into the Provider ID field.
 - If the user accesses the Search page and cancels the Search, they are returned to the Maintain Non-WISACWIS Benefits page.
 - The user can select null values in the Service Category and Service Type fields.
 - The user can write-in the Facility Name and Service Type in the two free-text fields provided.
 - One of the following fields are mandatory– Service Type or WI Service Type.
 - If both fields are null, pop-up the following error message – “Please enter a Service Type”.
 - One of the following fields are mandatory – Facility Name or WI Facility Name.
 - If both fields are null, pop-up for following error message – “Please enter a Facility Name”.
 - The Calculate button is enabled only when the Refund Amount is inserted and the provider is a WiSACWIS provider. It is not enabled when the provider is user-entered.
 - When the user selects the ‘Calculate’ button, the Maintenance amount and Title XIX amount are calculated in the following manner –
 - To select the appropriate FFP percentage:
 - Select IVE_FFP.am_ffp where
 - NON_SACWIS_BENEFITS.dt_appld >= IVE_FFP.dt_eff AND
 - IVE_FFP row has max(ts_cr).

- To select appropriate Provider Eligibility Cost row -
 - Select am_maint and am_titlexix where -
 - NON_SACWIS_BENEFITS.id_prvd_org = PRVD_ELIG_COST_AUDIT.id_prvd_org AND
 - NON_SACWIS_BENEFITS.dt_appld (MM/DD/YYYY) = PRVD_ELIG_COST_AUDIT.dt_eff (MM/YYYY) (Use only the year for comparison) AND
 - NON_SACWIS_BENEFITS.cd_srvc = SERVICE_TYPE.cd_srvc where SERVICE_TYPE.cd_sw_rptg_grp = PRVD_ELIG_COST_AUDIT.cd_sw_rpt_grp.
- Use am_maint and am_titlexix rates from the retrieved PRVD_ELIG_COST_AUDIT row with max(ts_cr).
- If no row found in PRVD_ELIG_COST_AUDIT, select am_maint and am_titlexix from -
 - NON_SACWIS_BENEFITS.id_prvd_org = PRVD_ELIG_COST.id_prvd_org AND
 - NON_SACWIS_BENEFITS.cd_srvc = PRVD_ELIG_COST.cd_srvc AND
 - NON_SACWIS_BENEFITS.dt_appld >= PRVD_ELIG_COST.dt_eff
 - Use PRVD_ELIG_COST am_maint and am_titlexix values from row with max(ts_cr).
- For Maint. Amt –
 - Check if FFP status of the child is ER (NON_SACWIS_BENEFITS = 'ER')
 - If FFP Status is <> 'ER'
 - Do nothing.
 - If FFP Status = 'ER' AND Facility Type = 'Foster Care'
 - Maint. Amt = Refund Amt (NON_SACWIS_BENEFITS.amt_rcvd) * IVE_FFP.am_ffp (selected from above).
 - If FFP Status = 'ER' AND Facility Type = 'Group Home'
 - Maint. Amt = Refund Amt (NON_SACWIS_BENEFITS.amt_rcvd) * IVE_FFP.am_ffp (selected from above) * PRVD_ELIG_COST_AUDIT.am_maint or PRVD_ELIG_COST.am_maint (selected from above).
 - If FFP Status = 'ER' AND Facility Type = 'RCC'
 - Maint. Amt = Refund Amt (NON_SACWIS_BENEFITS.amt_rcvd) * IVE_FFP.am_ffp (selected from above) * PRVD_ELIG_COST_AUDIT .am_maint or PRVD_ELIG_COST.am_maint (selected from above).
- For calculation of Title XIX Amt. –
 - Check if Facility Type is 'RCC'.

- If Facility Type <> 'RCC'
 - Do nothing.
- If Facility Type = 'RCC'
 - Title XIX Amt. = Refund Amt (NON_SACWIS_BENEFITS.amt_rcvd) * IVE_FFP.am_ffp (selected from above) * PRVD_ELIG_COST_AUDIT .am_titlexix or PRVD_ELIG_COST.am_titlexix (selected from above).

Save Processing:

- The user will be able to make changes to the record until the time of processing.
- Once the record is saved, the information is frozen and the user will not be able to make changes to the record.
- The data is saved in the NON_SACWIS_BENEFITS table.

If the user chooses not to save the data, they are returned to the main page with the information not being saved.

1.2.11.3.CRUD Matrix

| Table Name | CRUD |
|----------------------|-------------|
| NON SACWIS BENEFITS | CRUD |
| PROVIDER | R |
| PRVD ELIG COST | R |
| IVE_FFP | R |
| PRVD_ELIG_COST_AUDIT | R |
| SERVICE_TYPE | R |

Inventories

1.2.12. Table Descriptions

| Table Name | Description |
|-------------------|--|
| ADDRESS | This table maintains all ADDRESS information pertaining to a PERSON, except Intake Report (Report and Referral). Address detail consists of home, business and any other address that is identified as required for WiSACWIS processing. Processes of CM01 (Manage Person) build and maintain this information. |
| ASSIGNMENT | The ASSIGNMENT table maintains information pertaining to the ASSIGNMENT of a WORKER to a designated activity or a dimension of work for which a worker is held responsible. Category, type, role, and responsibility define the structure of worker assignment. ASSIGNMENTs can be associated with a CASE which may carry an associated weight, or they can be CASE, PROVIDER, or other work-related ASSIGNMENTs that do not carry any associated weight that feeds the measurement of workload capacity. Processes of CM02 (Maintain Worker Assignment) create and update this information. |
| BANK | The BANK table is used to maintain account information for each county's banks where trust account money is kept. Processes of FM07 (Trust Account) maintain this information. |
| BENEFIT | The BENEFIT table keeps a history and shows the current status of each of the benefits a child is expected to receive. Processes within FM07 (Trust Account) maintain this information. |
| CASE | A CASE is the focus of every human services business activity. A case is created as a result of intake work and can be one of three types: individual (which includes TPR and Adoption cases), family, and DHFS provider or other provider. The CASE table contains case relations, address and approval information. Processes of SM04 (Maintain Case) and SM05 (Close Case) create and maintain case records. |
| CODE DESC | The CODE DESC table is used to house the description fields for most of the reference data used in WiSACWIS. |
| ELIGIBILITY | The ELIGIBILITY table maintains information pertaining to a foster child's initial Title IV-E eligibility determination. Once the determination is complete, the child will be assigned an |

| | |
|------------------------|--|
| | eligibility status. Processes of FM03 (Eligibility) build and maintain this information. |
| ELIG REDET | The ELIG_REDET table maintains information pertaining to a foster child's Title IV-E eligibility re-determinations. Once the re-determination is complete, the child will be assigned an eligibility status. Processes of FM03 (Eligibility) build and maintain this information. |
| MAX BALANCE | The MAX_BALANCE table is used by the Trust Account Manager to maintain maximum balance amount for children with trust accounts. Processes within FM07 (Trust Account) maintain this information. |
| NON_SACWIS_BENEFITS | The NON_SACWIS_BENEFITS table is used by the fiscal and trust account staff to record trust account monies received by children who are not entered in WiSACWIS. Information accessed by BFS and fiscal staff. All processes maintained within FM07 (Trust Account). |
| PERSON | This PERSON table maintains information that identifies an individual known to DHFS or the county child welfare division such as name, date of birth, social security number, race, sex, etc. A PERSON can be a WORKER, REPORT PART, REFERRAL PART, CASE PART or PROVIDER PART. Primary search processing is centered around this data. Processes of CM01 (Manage Person) build and maintain this information. |
| TICKLER | Ticklers are an electronic notification to WiSACWIS workers of tasks to be completed by a certain time. The TICKLER table maintains information about all ticklers and alerts in WiSACWIS. CM04 (Ticklers) creates tickler information. |
| TRUST ACCOUNT | The TRUST_ACCOUNT table maintains a total of the child's savings balance and personal needs balance. Processes within FM07 (Trust Account) maintain this information. |
| TRUST ACCOUNT INTEREST | The TRUST_ACCOUNT_INTEREST table is used by the Trust Account Manager to post accrued interest to the trust accounts for each child. Processes within FM07 (Trust Account) maintain this information. |
| TRUST ACCOUNT LEDGER | The TRUST_ACCOUNT_LEDGER table is used by the Trust Account Manager to maintain a complete history of all debit and credit transactions for a particular trust account. Processes within FM07 (Trust Account) maintain this information. |
| WORKER | The WORKER table maintains information pertaining to an individual (PERSON) who is employed by a county or DHFS |

| | |
|--|---|
| | and is in a job class that provides services and/or a job class that receives eWiSACWIS-defined work assignments. The information is created in CM18 (Manage Worker). |
|--|---|

1.2.13. Reference Data

1.2.13.1.Drop Downs

Field Name: Type (Maintain Benefit Record)

Table Name: CODE_DESC_FNCL

Group ID: BENFTYPE

Field Name: Benefit is available for...

Table Name: CODE_DESC

Group ID: TRUSTEXP

Field Name: Type (Maintain Ledger Record)

Table Name: CODE_DESC_FNCL

Group ID: TRANTYPE

Field Name: Detail

Table Name: CODE_DESC_FNCL

Group ID: TRANDETL

Field Name: Benefit Type

Table Name: CODE_DESC_FNCL

Group ID: BENFTYPE

1.2.13.2.List Boxes

None

1.2.14. Automated Messages

None

1.2.15. Checklists

None

1.2.16. Ticklers

1.2.16.1. Spend Down Needed for FFP child

Category: Trust Account Ledger Tab, Trust Account Page

Type: Trust Account Spend Down

Description: Whenever a child balance reaches a specified percentage of the maximum balance, a tickler will be issued.

Creation: The tickler is created on the Ledger tab of the Trust Account page when a deposit is made and conditions hold. The conditions consist of when the balance of the child's trust account is equal the percentage times the maximum balance amount indicated in the Maximum Balance page. A maximum balance can be established for FFP and non-FFP children. A child must have eligibility rows in order for the tickler to be issued.

Upon creation of the tickler, a check for a County specific tickler is made. If a unique record exists for a specified County, and that County has decided to utilize this tickler, then a tickler is created with due dates, reminder dates, and escalation dates denoted in the CATEGORY_TYPE table. See CATEGORY_TYPE for County-specific tickler values.

If a unique record exists for a specified County, and that County has decided to not use this tickler, then no tickler is created for the County. See CATEGORY_TYPE table for County-specific information.

If a unique record does not exist for a specified County, then the base values for the Spend Down needed for FFP child tickler are utilized. The due dates, reminder dates, and escalation dates are then based on the values described below.

Deletion: When the withdrawal is made and the balance of the trust account goes below the maximum balance amount.

Due Date: The day the tickler was created

Reminder Date: Immediately

First Escalation Date: 10 days after the tickler is due

Second Escalation Date: 25 days after the tickler is due

1.2.16.2. Spend Down Needed for Non-FFP child

Category: Trust Account Ledger Tab, Trust Account Page

Type: Trust Account Spend Down

Description: Whenever a child balance reaches a specified percentage of the maximum balance, a tickler will be issued.

Creation: The tickler is created occurs on the Ledger tab of the Trust Account page when a deposit is made and conditions hold. The conditions consist of when the balance of the child's trust account is equal the percentage times the maximum balance amount indicated in the Maximum Balance page. A maximum balance can be established for FFP and non-FFP children. A child must have eligibility rows in order for the tickler to be issued.

Upon creation of the tickler, a check for a County specific tickler is made. If a unique record exists for a specified County, and that County has decided to utilize this tickler, then a tickler is created with due dates, reminder dates, and escalation dates denoted in the CATEGORY_TYPE table. See CATEGORY_TYPE for County-specific tickler values.

If a unique record exists for a specified County, and that County has decided to not use this tickler, then no tickler is created for the County. See CATEGORY_TYPE table for County-specific information.

If a unique record does not exist for a specified County, then the base values for the Spend Down needed for non-FFP child tickler are utilized. The due dates, reminder dates, and escalation dates are then based on the values described below.

Deletion: When the withdrawal is made and the balance of the trust account goes below the maximum balance amount.

Due Date: The day the tickler was created

Reminder Date: Immediately

First Escalation Date: 10 days after the tickler is due

Second Escalation Date: 25 days after the tickler is due

1.2.17. Notifications

None

1.2.18. Text Templates

None

1.2.19. Reports

1.2.19.1. FM07-01b Trust Account Current Balance Report

1.2.19.2. FM07-02 Trust Account Reconciliation Report

1.2.19.3. FM07-03b Placement Ending Report

1.2.19.4. FM07-04b Deposit History Report

1.2.19.5. FM07-05b Trust Account Balances for Closed Cases

1.2.19.6. FM07-06b Trust Account Drawdowns

1.2.20. Triggers

1.2.20.1. Online Trigger Txn Code 100

Description: An online trigger needs to be saved to the ONLINE_TRIGGER Table whenever there is a new record saved or updated to the Assets tab. This tells the system that the new information needs to be transmitted to Kids Information Data System (KIDS). The KIDS system is maintained by the Child Support Bureau within the Department of Workforce Development.

Perform a lookup with id_prsn to KIDS_TRANS, and create one trigger for each row where cd_kids = 2, only if no ONLINE_TRIGGER exists for this cd_txn and id_trigger2 = KIDS_TRANS.id_epsd and this id_prsn. (See topic paper *SM013b Assets and Employment*)

1.2.20.2. Online Trigger Txn Code 101

Description: An online trigger needs to be saved to the ONLINE_TRIGGER Table whenever there is a new record saved or updated to the Employment tab. This tells the system that the new information needs to be transmitted to

Kids Information Data System (KIDS). The KIDS system is maintained by the Child Support Bureau within the Department of Workforce Development.

Perform a lookup with id_prsn to KIDS_TRANS, and create one trigger for each row where cd_kids = 2, only if no ONLINE_TRIGGER exists for this cd_txn and id_trigger2 = KIDS_TRANS.id_epsd and this id_prsn.

1.2.20.3. Online Trigger Txn Code 103

Description: An online trigger needs to be saved to ONLINE_TRIGGER table, whenever there is a new record saved or updates to the following columns used on the Person Management page:

| | |
|-------------------|--------------------|
| CD_MTHRS_CRNT_STS | CD_FTHRS_CRNT_STS |
| NM_PRFX | NM_SFX |
| NM_FRST | NM_LST |
| CD_GNDR | DT_BRTH |
| TX_BRTH_PLC | DT_DTH |
| CD_STATE_RSDNT | CD_LNG_PRFR |
| CD_RLGN | CD_RACE |
| CD_RACE_TWO | CD_RACE_THREE |
| CD_CMBN_ETHN | CD_HSPNC |
| CD_INDN | CD_INDN2 |
| ID_CENSUS_NBR | FL_TEEN_PRNT |
| FL_RESIDES_HM_HR | FL_RCV_KNSHP_PYMNT |
| ID_CARES_PIN | NM_MDL_INTL |
| ID_MOM | ID_DAD |
| CD_MRTL_STAT | ID_SSN |

Perform a lookup with id_prsn to KIDS_TRANS, and only create a trigger if at least one has KIDS_TRANS.cd_kids = 2 (Note: there probably will be multiple rows retrieve when KIDS_TRANS lookup is performed, but only one row can have cd_kids = 2). Database Trigger. (See also *CM01 Person Management*)

1.2.20.4. Online Trigger Txn Code 105

Description: An online trigger needs to be saved to ONLINE_TRIGGER table, whenever there is a new record saved or updated to the Person Management page whenever a change occurs on the Address tab [any change or addition]. Perform a lookup with id_prsn to KIDS_TRANS, and only create a trigger if at least one has KIDS_TRANS.cd_kids = 2 (Note: there probably will be multiple

rows retrieve when EPISODE lookup is performed, but only one update). Database Trigger. (See also *CM01 Person Management*).

1.2.20.5. Online Trigger Txn Code 106

Description: Created when a new record is saved or updated to the Maintain Physical Address page. Use id_prvd_org to lookup EPISODE.id_prvd_org and if at least one of the select rows has cd_kids = 2, then issue the trigger. The triggers is processed by the KIDS Daily batch (See Topic *PM08 – Maintain Address*).

1.3. Batch Programs

1.3.1. Spread Interest Batch

Program Name: b-fm0701-spread-interest-batch

Script Name: b-fm07-00.script

Process Summary: The trust account administrator will receive on a monthly basis the interest accrued on the account held at FIRSTAR bank. When a statement of interest is received, the administrator will enter that amount into eWiSACWIS along with the date. The spread interest batch will read this amount and spread the amount evenly across each child's personal account. The following calculation is used, and the batch will perform the calculation based on the records up to the statement's date.

$$\frac{\sum_{\text{statement date}} \text{of child's ledger deposits} - \sum_{\text{statement date}} \text{of child's ledger withdrawals}}{\sum_{\text{statement date}} \text{of all ledger deposits} - \sum_{\text{statement date}} \text{of all ledger withdrawals}} = \text{child's percentage}$$

$$(\text{child's percentage})(\text{interest this month} \pm \text{any remainder from prior months}) = \text{child's portion of interest}$$

The calculated child's portion of interest will then be posted to the ledger record. The child percentage is calculated to the sixth decimal place (ex, 0.104731, i.e. 10.4731%). Because of rounding of dollar amounts when posting ledger deposits, there will usually be a small remainder that is indivisible. For example, the remainder \$0.02 cannot be divided among 400 accounts. This remainder will be stored in the Interest Table and applied towards the next interest batch run. In mathematical notation:

Upper Bound Interest Created Amount = (N) 0.005
Lower Bound Interest Lost Amount = (N) 0.0049999

Where, N = Number of Trust Accounts

- Frequency:** Monthly.
- Dependencies:** This batch program needs to run after FM07 – KIDS-Interface and FM07-SSI-Interface and FM07 – deposit batch
- Input Files:** cycle-date-override
- Output Files:** None
- Database Tables:**
- | | |
|----------------------|------------|
| TRUST_ACCOUNT | R |
| TRUST_ACCOUNT_LEDGER | C, R, U, D |

Process Description:

1. Select an interest record for the TRUST_ACCOUNT_INTEREST table if the fl_pstd = N. If 'Y' the interest for this record has already been spread.
2. Check to see if any records exist for
TRUST_ACCOUNT_LEDGER.cd_detl = 8 and
TRUST_ACCOUNT_LEDGER.dt_trans =
TRUST_ACCOUNT_INTEREST.dt_effctv and
using TRUST_ACCOUNT_LEDGER.id_trust_acct
select row TRUST_ACCOUNT.id_trust_acct and check
TRUST_ACCOUNT.id_bank =
TRUST_ACCOUNT_INTEREST.id_bank
for the record selected in step 1. If true, go to Section II. If False, continue with Section I.

3. Once Section II or Section I is complete for id_trust_acct with that id_bank, then calculate WS-INTEREST-AMT minus WS-TOTAL-DEPOSIT = INTEREST.am_rmndr for the row selected.
4. Go back to Step 1 and retrieve the next record with that condition.

Section I

5. Calculate WS-INTEREST-AMT = TRUST_ACCOUNT_INTEREST.am_intrst + TRUST_ACCOUNT_INTEREST.am_rmndr (Sum all rows TRUST_ACCOUNT_INTEREST.dt_effctv < TRUST_ACCOUNT_INTEREST.dt_effctv of row select in step 1)
6. With the first selected record perform the following calculation:
 - Sum all TRUST_ACCOUNT_LEDGER.am_trans records for which TRUST_ACCOUNT_LEDGER.dt_trans =< TRUST_ACCOUNT_INTEREST.dt_effctv and
 - using TRUST_ACCOUNT_LEDGER.id_trust_acct
 - select row TRUST_ACCOUNT.id_trust_acct and check TRUST_ACCOUNT.id_bank = TRUST_ACCOUNT_INTEREST.id_bank and TRUST_ACCOUNT and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1. Set this = WS-DEPOSIT-ALL
 - Sum all TRUST_ACCOUNT_LEDGER records for which TRUST_ACCOUNT_LEDGER.dt_trans =< TRUST_ACCOUNT_INTEREST.dt_effctv and
 - using TRUST_ACCOUNT_LEDGER.id_trust_acct
 - select row TRUST_ACCOUNT.id_trust_acct and check TRUST_ACCOUNT.id_bank = TRUST_ACCOUNT_INTEREST.id_bank and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2. Set this = WS-WITHDRAWAL-ALL
5. Select a unique TRUST_ACCOUNT_LEDGER.id_trust_acct and perform the following calculation for each id_trust_acct.
 - Sum all TRUST_ACCOUNT_LEDGER.am_trans records for which TRUST_ACCOUNT_LEDGER.dt_trans =< TRUST_ACCOUNT_INTEREST.dt_effctv and

TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1 and
TRUST_ACCOUNT_LEDGER.id_trust_acct = unique and
using TRUST_ACCOUNT_LEDGER.id_trust_acct
select row TRUST_ACCOUNT.id_trust_acct and check
TRUST_ACCOUNT.id_bank =
TRUST_ACCOUNT_INTEREST.id_bank and
for this TRUST_ACCOUNT.id_trust_acct, TRUST_ACCOUNT.dt_closed = null
and set this = WS-DEPOSIT-CHILD

- Sum all TRUST_ACCOUNT_LEDGER.am_trans records for which
TRUST_ACCOUNT_LEDGER.dt_trans =<
TRUST_ACCOUNT_INTEREST.dt_effctv and
TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2 and
TRUST_ACCOUNT_LEDGER.id_trust_acct = unique and
using TRUST_ACCOUNT_LEDGER.id_trust_acct
select row TRUST_ACCOUNT.id_trust_acct and check
TRUST_ACCOUNT.id_bank =
TRUST_ACCOUNT_INTEREST.id_bank and
for this TRUST_ACCOUNT.id_trust_acct, TRUST_ACCOUNT.dt_closed = null
and set this = WS-WITHDRAWAL-CHILD
- If WS-DEPOSIT-CHILD – WS-WITHDRAWAL-CHILD =< 0, stop processing
records for this id_prsn, else
- Divide (WS-DEPOSIT-CHILD – WS-WITHDRAWAL-CHILD) / (WS-DEPOSIT-
ALL – WS-WITHDRAWAL-ALL) = WS-PERCENTAGE
- Multiply WS-PERCENTAGE(WS-INTEREST-AMT) = WS-CHILD-PORITION
- Post WS-CHILD-PORITION to the TRUST_ACCOUNT_LEDGER.am_trans
- Add WS-CHILD-PORITION to WS-TOTAL-DEPOSITS
- Post “Y” to the TRUST_ACCOUNT_INTEREST.fl_pstd
- Post the TRUST_ACCOUNT_LEDGER.id_case of id_prsn select in step 4
- Post the TRUST_ACCOUNT_LEDGER.id_trust_acct selected in step 4
- Post the a unique number sequential number for the
TRUST_ACCOUNT_LEDGER.id_ldgr
- Post null to the TRUST_ACCOUNT_LEDGER.id_chk_nmbr
- Post “1” to TRUST_ACCOUNT_LEDGER.cd_trans
- Post “8” to the TRUST_ACCOUNT_LEDGER.cd_detl

- Post “10” to the TRUST_ACCOUNT_LEDGER.cd_bnft_typ
- Post text message “Monthly Interest Deposit” to the TRUST_ACCOUNT_LEDGER.tx_desc
- Post “9999” to the TRUST_ACCOUNT_LEDGER.id_cr
- Post current system time to the TRUST_ACCOUNT_LEDGER.ts_cr

Section II

7. Calculate WS-INTEREST-AMT = TRUST_ACCOUNT_INTEREST.am_intrst + TRUST_ACCOUNT_INTEREST.am_rmndr (Sum all rows TRUST_ACCOUNT_INTEREST.dt_effctv < TRUST_ACCOUNT_INTEREST.dt_effctv of row select in step 1)
8. Go through and delete all records for which:
TRUST_ACCOUNT_LEDGER.cd_detl = 8 and
TRUST_ACCOUNT_LEDGER.dt_trans =
TRUST_ACCOUNT_INTEREST.dt_effctv for the record selected in step 1.
9. With the first selected record perform the following calculation:
 - Sum all TRUST_ACCOUNT_LEDGER.am_trans records for which
TRUST_ACCOUNT_LEDGER.dt_trans =<
TRUST_ACCOUNT_INTEREST.dt_effctv and
using TRUST_ACCOUNT_LEDGER.id_trust_acct
select row TRUST_ACCOUNT.id_trust_acct and check
TRUST_ACCOUNT.id_bank =
TRUST_ACCOUNT_INTEREST.id_bank and
TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1. Set this = WS-DEPOSIT-ALL
 - Sum all TRUST_ACCOUNT_LEDGER.am_trans records for which
TRUST_ACCOUNT_LEDGER.dt_trans =<
TRUST_ACCOUNT_INTEREST.dt_effctv and
using TRUST_ACCOUNT_LEDGER.id_trust_acct
select row TRUST_ACCOUNT.id_trust_acct and check
TRUST_ACCOUNT.id_bank =
TRUST_ACCOUNT_INTEREST.id_bank and
TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2. Set this = WS-WITHDRAWAL-ALL

7. Select a unique TRUST_ACCOUNT_LEDGER.id_trust_acct and perform the following calculation for each id_trust_acct.

- Sum all TRUST_ACCOUNT_LEDGER records for which TRUST_ACCOUNT_LEDGER.dt_trans =< TRUST_ACCOUNT_INTEREST.dt_effctv and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1 and
using TRUST_ACCOUNT_LEDGER.id_trust_acct
select row TRUST_ACCOUNT.id_trust_acct and check
TRUST_ACCOUNT.id_bank =
TRUST_ACCOUNT_INTEREST.id_bank and
for this TRUST_ACCOUNT.id_trust_acct, TRUST_ACCOUNT.dt_closed = null
TRUST_ACCOUNT_LEDGER.id_prsn = unique. Set this = WS-DEPOSIT-CHILD
- Sum all TRUST_ACCOUNT_LEDGER records for which TRUST_ACCOUNT_LEDGER.dt_trans =< TRUST_ACCOUNT_INTEREST.dt_effctv and
using TRUST_ACCOUNT_LEDGER.id_trust_acct
select row TRUST_ACCOUNT.id_trust_acct and check
TRUST_ACCOUNT.id_bank =
TRUST_ACCOUNT_INTEREST.id_bank and
for this TRUST_ACCOUNT.id_trust_acct, TRUST_ACCOUNT.dt_closed = null
TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2 and
TRUST_ACCOUNT_LEDGER.id_prsn = unique. Set this = WS-WITHDRAWAL-CHILD
- If WS-DEPOSIT-CHILD – WS-WITHDRAWAL-CHILD =< 0, stop processing records for this id_prsn (i.e. no interest for this person will be posted), else
- Divide (WS-DEPOSIT-CHILD – WS-WITHDRAWAL-CHILD) / (WS-DEPOSIT-ALL – WS-WITHDRAWAL-ALL) = WS-PERCENTAGE
- Multiply WS-PERCENTAGE(TRUST_ACCOUNT_INTEREST.am_intrst) = WS-CHILD-PORTION
- Post WS-CHILD-PORTION to the TRUST_ACCOUNT_LEDGER.am_trans
- Add WS-CHILD-PORTION to WS-TOTAL-DEPOSITS
- Post “Y” to the TRUST_ACCOUNT_INTEREST.fl_pstd
- Post the TRUST_ACCOUNT_LEDGER.id_case of id_prsn select in step 4
- Post the a unique number sequential number for the TRUST_ACCOUNT_LEDGER.id_ldgr

- Post the TRUST_ACCOUNT_LEDGER.id_trust_acct selected in step 4
- Post null to the TRUST_ACCOUNT_LEDGER.id_chk_nmbr
- Post “1” to TRUST_ACCOUNT_LEDGER.cd_trans
- Post “8” to the TRUST_ACCOUNT_LEDGER.cd_detl
- Post “10” to the TRUST_ACCOUNT_LEDGER.cd_bnft_typ
- Post text message “Monthly Interest Deposit” to the TRUST_ACCOUNT_LEDGER.tx_desc
- Post “9999” to the TRUST_ACCOUNT_LEDGER.id_cr
- Post current system time to the TRUST_ACCOUNT_LEDGER.ts_cr
- Post “N” to the TRUST_ACCOUNT_LEDGER.fl_reversal

1.3.2. Trust Account Deposit Batch

Program Name: b-fm0700-deposit-batch

Script Name: b-fm07-00.script

Process Summary: The batch will read the Benefit record table and post values to the Ledger table. The Benefit Record is a listing of the monthly payments a child is expected to receive, from some or all of the following: SSI, SSA, VA, Railroad Retirees Benefits or IVD. Once a record is selected, a personal-needs deduction is made against this amount and both the benefit amount and the personal needs amount are posted to the Ledger table.

Frequency: Daily

Dependencies: This batch program needs to run after FM07 – KIDS-Interface and FM07-SSI-Interface

Input Files: cycle-date-override

Output Files: None

Database Tables:

| | |
|----------------------|---------|
| BENEFIT | R, U |
| TRUST_ACCOUNT_LEDGER | C, R, U |

Process Description:

1. Select a benefit record from the BENEFIT table and check to see if BENEFIT.cd_eft = 0,1,2

- If `cd_eft = 0` then check to see if `BENEFIT.dt_pstd_lst = or < parmdate - number of days given in BENEFIT.am_days`. Or `BENEFIT.am_days = null`, then `parmdate - one month` (e.g. `9/10/99 - 1/0/0 = 8/10/1999`). Select record if meets criteria.
 - If `cd_eft = 0` then check to see if `BENEFIT.dt_pstd_lst = null` and `BENEFIT.dt_efctv =< parmdate`, then select record.
 - If `cd_eft = 1` then do not select record
 - If `cd_eft = 2` then do not select record
2. Also, check to see if `BENEFIT.dt_effctv =< parmdate`, do not select if criteria not met. Do not select record if `BENEFIT.dt_clsng =< parmdate`.

Section I

3. With selected `BENEFIT.am_bnft - BENEFIT.am_prsnl_nds = ws-trans-amount`. If `ws-trans-amount < 0`, then set `ws-trans-amount = 0` and in step 8 post `BENEFIT.am_bnft` (instead of `am_prsnl_nds`) to `TRUST_ACCOUNT_LEDGER.am_trans` and continue with step 8.
4. Post `ws-trans-amount` to the `TRUST_ACCOUNT_LEDGER.am_trans`. Also populate:
- `TRUST_ACCOUNT_LEDGER.id_case = BENEFIT.id_case`
 - `TRUST_ACCOUNT_LEDGER.id_prsn = BENEFIT.id_prsn`
 - `TRUST_ACCOUNT_LEDGER.id_trust_acct = BENEFIT.id_trust_acct`
 - `TRUST_ACCOUNT_LEDGER.id_ldgr = a unique sequential number`
 - `TRUST_ACCOUNT_LEDGER.id_chk_nmbr = null`
 - `TRUST_ACCOUNT_LEDGER.dt_trans = current parmdate`
 - `TRUST_ACCOUNT_LEDGER.cd_trans_typ = '1'` (i.e. Deposit)
 - `TRUST_ACCOUNT_LEDGER.id_cr = "8181"`
 - `TRUST_ACCOUNT_LEDGER.ts_cr = current system date`
4. Post `BENEFIT.am_prsnl_nds` to the `TRUST_ACCOUNT_LEDGER.am_trans`, but do not post a record if `BENEFIT.am_prsnl_nds = 0`. If a record is created, also populate:
- `TRUST_ACCOUNT_LEDGER.id_case = BENEFIT.id_case`
 - `TRUST_ACCOUNT_LEDGER.id_prsn = BENEFIT.id_prsn`
 - `TRUST_ACCOUNT_LEDGER.id_trust_acct = BENEFIT.id_trust_acct`
 - `TRUST_ACCOUNT_LEDGER.id_ldgr = a unique sequential number`

- TRUST_ACCOUNT_LEDGER.id_chk_nmbr = null
 - TRUST_ACCOUNT_LEDGER.dt_trans = current parmdate
 - TRUST_ACCOUNT_LEDGER.cd_trans_typ = '1' (i.e. Deposit)
 - TRUST_ACCOUNT_LEDGER.id_cr = "8181"
 - TRUST_ACCOUNT_LEDGER.ts_cr = current system date
5. When a benefit type is posted to the TRUST_ACCOUNT_LEDGER, post for that benefit type the current date to the BENEFIT.dt_pstd_lst field, and if BENEFIT.cd_eft = 2, update BENEFIT.cd_eft = 1
6. Evaluate BENEFIT.cd_bnft_typ:
- If 1, move 1 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY SSI STATE -- DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 2, move 2 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY SSI FEDERAL -- DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 3, move 3 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY SSA DISABILITY DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 4, move 4 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY SSA SURVIVOR'S DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 5, move 5 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY IV-D DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 6, move 6 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY VA DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 7, move 7 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY RAILROAD RETIREES DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 8, move 8 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY PERSONAL NEEDS DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
 - If 9, move 9 to TRUST_ACCOUNT_LEDGER.cd_bnft_typ and post text message "MONTHLY OTHER BENEFITS DEPOSIT" to TRUST_ACCOUNT_LEDGER.tx_desc
7. Post to the TRUST_ACCOUNT_LEDGER.cd_detl = (value of Deposit Batch, from CODE_DESC_FNCL.id_fncl_grp = TRANDETL)

8. For those records on the BENEFIT table that were selected, post the parmdate into the BENEFIT.dt_pstd_lst
9. Go through and calculate balance, for those people whose TRUST_ACCOUNT.id_tcklr = null.
 - Using the person TRUST_ACCOUNT.id_prsn, sum records TRUST_ACCOUNT_LEDGER.am_trans for TRUST_ACCOUNT_LEDGER.id_prsn and TRUST_ACCOUNT_LEDGER.id_trust_acct TRUST_ACCOUNT.id_trust_acct and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1, Set this = WS-DEPOSIT-ALL
 - Using the person TRUST_ACCOUNT.id_prsn, sum records TRUST_ACCOUNT_LEDGER.am_trans for TRUST_ACCOUNT_LEDGER.id_prsn and TRUST_ACCOUNT_LEDGER.id_trust_acct TRUST_ACCOUNT.id_trust_acct and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2, Set this = WS-WITHDRAWAL-ALL
 - Calculate WS-DEPOSIT-ALL minus WS-WITHDRAWAL-ALL = WS-CURRENT-BALANCE
10. If child is FFP (retrieve most recent record ELIG_REDET.cd_stat = 1 and ELIG_REDET.fl_cmplt = Y and ELIG_REDET.fl_void <> Y. (Or if no record on ELIG_REDET, then ELIGIBILITY.cd_stat = 1 and ELIGIBILITY.fl_cmplt = Y and ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and find most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp = "N", and calculate MAX_BALANCE.am_pcntg multiplied by MAX_BALANCE.am_maxbal = WS-MAX-BAL-NOTIFY. If WS-MAX-BAL-NOTIFY =< WS-CURRENT-BALANCE, then issue a tickler (go to step 12).
11. This step is only necessary if Step 11 indicates. Create a "Spend Down needed for a Non-FFP child" tickler. For this tickler set:
 - TICKLER.id_prsn = With TRUST_ACCOUNT.id_case lookup for ASSIGNMENT.id_grp_lvl1 and ASSIGNMENT.cd_asgn_role = '1' and ASSIGNMENT.dt_end = null. Use this ASSIGNMENT.id_prsn
 - TICKLER.id_tcklr = unique number
 - TICKLER.cd_ctgry = 11
 - TICKLER.cd_type = 1
 - TICKLER.cd_prmry_type = C
 - TICKLER.dt_due = parmdate

- TICKLER.id_prmry_type = ?????
 - TICKLER.tx_dscr = “Trust account for (insert BENEFIT.id_prsn) must be spent down by (insert parmdate + qt_escltn_spv)”
 - TICKLER.dt_escltn_spv = parmdate + CATEGORY_TYPE.qt_escltn_days_spv
 - TICKLER.dt_escltn_pspv = parmdate + CATEGORY_TYPE.qt_escltn_days_psv
 - TICKLER.dt_rmdr = parmdate + CATEGORY_TYPE.qt_lag_days
- TICKLER.id_cr = 8181
 - TICKLER.ts_cr = systemdate
 - TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr (from above)
12. If child is non-FFP, (ELIG_REDET.cd_stat not = 1 and ELIG_REDET.fl_cmplt = Y and ELIG_REDET.fl_void <> Y (or if no record on ELIG_REDET, then ELIGIBILITY.cd_stat not = 1 and ELIGIBILITY.fl_cmplt = Y and ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and find most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp = “Y”, and calculate MAX_BALANCE.am_non_ffp_pcntg multiplied by MAX_BALANCE.am_non_ffp_maxbal = WS-MAX-BAL-NOTIFY. If WS-FFP_MAX-BAL-NOTIFY =< WS-CURRENT-BALANCE , then issue a tickler (go to step 14).
13. This step is only necessary if Step 13 indicates. Create a “Spend Down needed for a Non-FFP child” tickler. For this tickler set:
- TICKLER.id_prsn = With TRUST_ACCOUNT.id_case lookup for ASSIGNMENT.id_grp_lv11 and ASSIGNMENT.cd_asgn_role = ‘1’ and ASSIGNMENT.dt_end = null. Use this ASSIGNMENT.id_prsn
 - TICKLER.id_tcklr = unique number
 - TICKLER.cd_ctgry = 11
 - TICKLER.cd_type = 2
 - TICKLER.cd_prmry_type = C
 - TICKLER.dt_due = parmdate
 - TICKLER.id_prmry_type = ?????
 - TICKLER.tx_dscr = “Trust account for (insert BENEFIT.id_prsn) must be spent down by (insert parmdate + qt_escltn_spv)”
 - TICKLER.dt_escltn_spv = parmdate + CATEGORY_TYPE.qt_escltn_days_spv

- TICKLER.dt_escltn_pspv = parmdate + CATEGORY_TYPE.qt_escltn_days_psv
 - TICKLER.dt_rmdr = parmdate + CATEGORY_TYPE.qt_lag_days
 - TICKLER.id_cr = 8181
 - TICKLER.ts_cr = systemdate
 - TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr (from above)
14. A tickler check must also be done, and a tickler should be removed if conditions apply. Select a record TRUST_ACCOUNT.id_tcklr not = null. If field is populated, then there is an outstanding tickler.
- Using the person TRUST_ACCOUNT.id_prsn, sum records TRUST_ACCOUNT_LEDGER.am_trans for TRUST_ACCOUNT_LEDGER.id_prsn and TRUST_ACCOUNT_LEDGER.id_trust_acct TRUST_ACCOUNT.id_trust_acct and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1, Set this = WS-DEPOSIT-ALL
 - Using the person TRUST_ACCOUNT.id_prsn, sum records TRUST_ACCOUNT_LEDGER.am_trans for TRUST_ACCOUNT_LEDGER.id_prsn and TRUST_ACCOUNT_LEDGER.id_trust_acct TRUST_ACCOUNT.id_trust_acct and TRUST_ACCOUNT_LEDGER.cd_trans_typ = 2, Set this = WS-WITHDRAWAL-ALL
 - Calculate WS-DEPOSIT-ALL minus WS-WITHDRAWAL-ALL = WS-CURRENT-BALANCE
 - If Non-FFP (ELIG_REDET.cd_stat not = 1 and ELIG_REDET.fl_cmplt = Y and ELIG_REDET.fl_void <> Y (or if no record on ELIG_REDET, then ELIGIBILITY.cd_stat not = 1 and ELIGIBILITY.fl_cmplt = Y and ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and find most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp = "Y", and calculate MAX_BALANCE.am_non_ffp_pcntg multiplied by MAX_BALANCE.am_non_ffp_maxbal = WS-MAX-BAL-NOTIFY. If WS-MAX-BAL-NOTIFY > WS-CURRENT-BALANCE, delete record for TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr, and move spaces to TRUST_ACCOUNT.id_tcklr
 - If FFP (retrieve most recent record ELIG_REDET.cd_stat = 1 and ELIG_REDET.fl_cmplt = Y and ELIG_REDET.fl_void <> Y. (Or if no record on ELIG_REDET, then ELIGIBILITY.cd_stat = 1 and ELIGIBILITY.fl_cmplt = Y and ELIGIBILITY.fl_void <> Y)), then lookup to MAX_BALANCE table and

find most recent MAX_BALANCE.dt_effctv and MAX_BALANCE.fl_non_ffp = "N", and calculate MAX_BALANCE.am_ffp_pcntg multiplied by MAX_BALANCE.am_ffp_maxbal = WS-MAX-BAL-NOTIFY. If WS-FFP_MAX-BAL-NOTIFY > WS-CURRENT-BALANCE , delete record for TRUST_ACCOUNT.id_tcklr = TICKLER.id_tcklr, and move spaces to TRUST_ACCOUNT.id_tcklr

1.3.3. KIDS to WiSACWIS Daily Batch

Topic: *FM07: KIDS to WiSACWIS Daily Update Batch*

Program Name: b-fm0705-KIDS-daily-update

Report Name: fm0705-b-daily-report (KIDS Data Compare Report)

Batch ID: 7450

Process Summary: This batch will produce a comparison report between WiSACWIS and KIDS Data. KIDS will send to WiSACWIS on a daily basis a flat file with records that have been changed in their system. These changed data elements include non-custodial address, name and amount of support. WiSACWIS will retrieve the appropriate case information and produce a comparison report.

Dependencies: None.

Frequency: Daily.

Runtime Parameters: None.

Selection Criteria: Use the case id to retrieve records

Sort Criteria: As the records appear in the flat file

Level Breaks:

- 1) Page break whenever a new id_case is discovered in the KIDS flat file. Every case will start a new a separate page.
- 2) Page break after first two addresses for an id_case are put on the report, and page break after every three addresses thereafter.

Input Data: cycle-date-override
fm0705inputfile

Output Data: This report lists the data changes that have occurred in KIDS and produces a comparison report with WiSACWIS data.

Audience: DHFS case maintenance staff.

Report Business Intent: 1) Confirm that the data that exists in KIDS matches WiSACWIS. 2) Update Trust Accounts with latest deposit information.

| | | |
|-------------------------|---------------|-------|
| Database Tables: | BENEFIT | C,R,U |
| | TRUST_ACCOUNT | R |
| | CODE_DESC | R |
| | PERSON | R |
| | ADDRESS | R |

Process Description:

There are two aspects to this batch. One produces a report and the second aspect updates trust account information.

- 1) Read through the flat file from KIDS (name TBD) and post data elements under the “KIDS Data” column of report. Perform the look up indicated to WiSACWIS and retrieve the associated data elements and display under the “WiSACIWS Data” column. If no data is retrieved from the flat file, then display “No Data Found.” If no data is retrieved from WiSACWIS, then display “No Data Found.”
- 2) For trigger 94, perform update indicated from the flat file to database as indicated to the BENEFIT table. This is the Trust Account financial information.

An incoming file will have the following layout:

Header (imbedded with trigger id) + Trigger

| KIDS data elements. Search and place data under KIDS column. | | Field Description | Row | WiSACWIS data location. Search and place data under WiSACWIS column if Row is specified. |
|--|--------------|----------------------------|--------------------|--|
| Position | Field Length | | | |
| 1 | 14 | Case ID | Case ID | In the KIDS header file, this will contain the equivalent id_case. Store in WS-CASE. |
| 2 | 4 | Trigger Number | Not displayed | This field describes one of six triggers which follow: 91, 92, 94, 95, 96, 98 |
| 3 | 8 | Date of Processing | Date of Processing | Date of processing |
| 4 | 10 | WiSACWIS Person ID | not displayed | In the KIDS header file, this will contain the equivalent id_prsn. Store in WS-PERSON . Perform lookup to KIDS_TRANS.id_prsn. If cd_role = C, set WS-ORD-SEQ = 000 If cd_role = M, set WS-ORD-SEQ = 001 If cd_role = D, set WS-ORD-SEQ = 002 |
| 5 | 3 | FIPS code: use field to | County | Use WS-PERSON and perform lookup to CASE and use cd_rgn and lookup to CODE_DESC.id_grp = REGION, |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|--|--------------|--|----------------|---|
| Position | Field Length | | | |
| | | lookup to LOCATION.cd_fips and retrieve tx_ofc_div | | cd_rgn = id_grpi |
| <p>Trigger 91 (Record Length 122 + Filler Length 0 = 122)</p> <p>Possible multiple rows. For every trigger 91, create a set of new Address rows on the Kids Data Compare report. Proceed each row by populating Person field, which will indicate either</p> <ol style="list-style-type: none"> 1) Child if Header pos8 = 000 2) Mother if Header pos8 = 001 3) Father if Header pos8 = 002 <p>Retrieve the associating Address in the WiSACWIS Data Column. This process is described below. The max number of Address would by 9 (cd_grp_type(s)) times 3 participants = 27</p> | | | | |
| 1 | 9 | SSN | Person's SSN | Use WS-PERSON and retrieve PERSON.id_ssn |
| 2 | 8 | Birthday | Birth Date | Use WS-PERSON and retrieve PERSON.dt_brth |
| 3 | 31 | Address 1 | Address Line 1 | Use WS-PERSON and lookup to ADDRESS.id_addrss_grp and retrieve |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|-------------------|----------------|---|
| Position | Field Length | | | |
| | | | | ADDRESS.ad_strt_nbr, and ad_strt_nme for ADDRESS.cd_grp = 1 and ADDRESS.cd_grp_type = (flat file pos. 10) |
| 4 | 31 | Address Line 2 | Address Line 2 | Use WS-PERSON and lookup to ADDRESS.id_prsn and retrieve ADDRESS.ad_ln2 for ADDRESS.cd_grp = 1 and ADDRESS.cd_grp_type = (flat file pos. 10) |
| 5 | 16 | City | City | Use WS-PERSON and lookup to ADDRESS.id_prsn and retrieve ADDRESS.tx_town for ADDRESS.cd_grp = 1 and ADDRESS.cd_grp_type = (flat file pos. 10) |
| 6 | 2 | State | State | Use WS-PERSON and lookup to ADDRESS.id_prsn and retrieve ADDRESS.cd_state for |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|-------------------|-----------|--|
| Position | Field Length | | | |
| | | | | ADDRESS.cd_grp = 1 and ADDRESS.cd_grp_type = (flat file pos. 10) |
| 7 | 9 | Zip | Zip | Use WS-PERSON and lookup to ADDRESS.id_prsn and retrieve ADDRESS.cd_zip for ADDRESS.cd_grp = 1 and ADDRESS.cd_grp_type = (flat file pos. 10) |
| 8 | 5 | Apartment | Apartment | Use WS-PERSON and lookup to ADDRESS.id_prsn and retrieve ADDRESS.ad_aprt for ADDRESS.cd_grp = 1 and ADDRESS.cd_grp_type = (flat file pos. 10) |
| 9 | 10 | Telephone | Telephone | Use WS-PERSON and lookup to ADDRESS.id_prsn and retrieve ADDRESS.ad_phone for ADDRESS.cd_grp = 1 and ADDRESS.cd_grp_type = (flat file pos. |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|-------------------|--------------|---|
| Position | Field Length | | | |
| | | | | 10) |
| 10N | 1 | Address Type | Address Type | <p>KIDS will send us indicator as to what kind of address they are sending:</p> <ul style="list-style-type: none"> 1 = Primary Residence 2 = Mailing 3 = Secondary Residence 5 = Responsible Person 6 = Last Known 7 = Parent Home Address 8 = Parent Not Residing in Household 9 = Work <p>Display the appropriate description above, by retrieving it from CODE_DESC.id_grp = PRSNADDR.</p> |
| <p>Trigger 94 Support Order Change Record Length 23 + Filler Length 99 = 122 (not used in WiSACWIS)</p> | | | | |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|----------------|---------------------------------|---------------|--|
| Position | Field Length | | | |
| 1 | 8 | Date Support Ordered (CCYYMMDD) | not displayed | not used |
| 2 | 10 [S9(08)V99] | Amount | not displayed | not used |
| 3 | 4 | Payment frequency | not displayed | not used |
| 4 | 1 | Payment made to clerk of court | not displayed | not used |
| 5 | 99 | filler | | |
| Trigger 95 (Record Length 71 + filler Length 51 = 122) | | | | |
| 1 | 20 | new last name | Last Name | use WS-PERSON and perform lookup to PERSON.nm_lst If WS-ORD-SEQ = 000 then post to Child 's Name If WS-ORD-SEQ = 001 then post to Mother's Name If WS-ORD-SEQ = 002 |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|-------------------|-------------|---|
| Position | Field Length | | | |
| | | | | then post to Father's Name |
| 2 | 15 | New first name | First Name | use WS-PERSON and perform lookup to PERSON.nm_frst If WS-ORD-SEQ = 000 then post to Child 's Name If WS-ORD-SEQ = 001 then post to Mother's Name If WS-ORD-SEQ = 002 then post to Father's Name |
| 3 | 15 | new middle name | Middle Name | use WS-PERSON and perform lookup to PERSON.nm_mdl_intl |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|--------------------|----------------|--|
| Position | Field Length | | | |
| 4 | 3 | child's new suffix | Child's Suffix | use WS-PERSON and perform lookup to PERSON.nm_sfx If WS-ORD-SEQ = 000 then post to Child 's Name If WS-ORD-SEQ = 001 then post to Mother's Name If WS-ORD-SEQ = 002 then post to Father's Name |
| 5 | 9 | Child's SSN | not used | use WS-PERSON and perform lookup to PERSON.cd_ssn If WS-ORD-SEQ = 000 then post to Child 's Name If WS-ORD-SEQ = 001 then post to Mother's Name If WS-ORD-SEQ = 002 then post to Father's Name |
| 6 | 8 | Date of Birth | not used | use WS-PERSON and perform lookup |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|-------------------|----------|--|
| Position | Field Length | | | |
| | | | | to PERSON.dt_brth If WS-ORD-SEQ = 000 then post to Child 's Name If WS-ORD-SEQ = 001 then post to Mother's Name If WS-ORD-SEQ = 002 then post to Father's Name |
| 7 | 1 | Ethnic Group | not used | use WS-PERSON and perform lookup to PERSON.cd_hspnc. If 'Y', then Hispanic. If <> 'Y' lookup to PERSON.cd_race then retrieve CODE_DESC.tx_desc_lrg for CODE_DESC.id_grp = PRSNRACE CODE_DESC.id_grpi = PERSON.cd_race If WS-ORD-SEQ = 000 then post to Child 's Name If WS-ORD-SEQ = 001 |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|----------------------------------|--|---|
| Position | Field Length | | | |
| | | | | then post to Mother's Name If WS-ORD-SEQ = 002 then post to Father's Name |
| 8 | 51 | filler | not displayed | |
| Trigger 96 (Record Length 22 + Filler Length 100 = 122) | | | | |
| 1 | 10 | Parent's WiSACWIS id_prsn | Mother's Person ID or Father's Person ID | Not used |
| 2 | 4 | Parent disposition code | Parent's Disposition | Not used |
| 3 | 8 | Parent Disposition Date | Disposition Date | not used |
| 13 | 36 | filler | | |
| Trigger 98 (Record Length 26 + Filler 96 = 122) | | | | |
| 1 | 10 | Kinship Caregiver's WiSACWIS Pin | not displayed | not used |
| 2 | 8 | Date of Non-Cooperation | not displayed | not used |

| <u>KIDS data elements.</u> Search and place data under KIDS column. | | Field Description | Row | <u>WiSACWIS data location.</u> Search and place data under WiSACWIS column if Row is specified. |
|---|--------------|-----------------------------|---------------|---|
| Position | Field Length | | | |
| | | Start | | |
| 3 | 8 | Date of Non-Cooperation End | not displayed | not used |
| 4 | 96 | filler | filler | |

```

1      10      20      30      40      50      60      70      80      90      100     110     120     130
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
Date: MM/DD/CCYY                Wisconsin Dept. of Health and Family Services                Report ID:
FM0705
Time: HH:MM PM                  Division of Child and Family Services                Page: 9,999
                                KIDS Data Compare Report

Site:           XXXXXXXXXXXXXXXXXXXXXXXX
Worker ID:      XXXXXXXXXXXX
Worker Name:    LXXXXXXXXXXXXXXXXXXXX, FXXXXXXXXXXXXXXXXXXXX, X
Worker Phone:  XXX-XXX-XXXX
Worker Fax:     XXX-XXX-XXXX
Date of Processing: XX/XX/XXXX

                                KIDS Data                                WiSACWIS Data
Case ID:        9999999999
Participant Name: LXXXXXXXXXXXXXXXXXXXX, FXXXXXXXXXXXXXXXXXXXX, X    LXXXXXXXXXXXXXXXXXXXX, FXXXXXXXXXXXXXXXXXXXX, X
Participant Role: XXXXXX
Participant ID:  XXXXXXXXXXXX
Participant SSN: XXXXXXXXXXXX
Participant DOB: XXXXXXXX
Participant Ethnicity: XXXXXXXXXXXXXXXXXXXX

Address 1
Person          LXXXXXXXXXXXXXXXXXXXX, FXXXXXXXXXXXXXXXXXXXX, X    LXXXXXXXXXXXXXXXXXXXX, FXXXXXXXXXXXXXXXXXXXX, X
Address Type:   XXXXXXXXXXXXXXXX
Address Line 1: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Address Line 2: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Apartment:     XXXXX
City. State, Zip: XXXXXXXXXXXXXXXXXXXXXXXX, XX, 99999    XXXXXXXXXXXXXXXXXXXXXXXX, XX, 99999
Telephone:     XXXXXXXXXXXX

Address 2
Person          LXXXXXXXXXXXXXXXXXXXX, FXXXXXXXXXXXXXXXXXXXX, X    LXXXXXXXXXXXXXXXXXXXX, FXXXXXXXXXXXXXXXXXXXX, X
Address Type:   XXXXXXXXXXXXXXXX
Address Line 1: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Address Line 2: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Apartment:     XXXXX
City. State, Zip: XXXXXXXXXXXXXXXXXXXXXXXX, XX, 99999    XXXXXXXXXXXXXXXXXXXXXXXX, XX, 99999
Telephone:     XXXXXXXXXXXX

1      10      20      30      40      50      60      70      80      90      100     110     120     130
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+

```


1.4.4 WiSACWIS to KIDS Daily Batch

Program Name: b-fm0706-kids-daily-extract

Script Name:

Process Summary: WiSACWIS will send on a daily basis case information to the KIDS system. Any change in WiSACWIS to Person Management, Assets and Employment, Provider Address, Worker Assignment will create a trigger to indicate that an update file must be sent to KIDS.

Frequency: Daily

Report Name: fm0706-b-exptn-report

Database Changes:

| Table Name | Column Name | Data Type | Width | Null | Key | Description |
|------------|-------------|-----------|-------|------|-----|---|
| EPISODE | cd_kids | FLOAT | | Yes | No | 1= Not Posted to KIDS 2= Posted to KIDS first time 3= Posted to KIDS as closed. Final cost of care payment sent. 4= All episodes closed. Costs of care fully reimbursed. 5= Conversion value. Ignored by all processing. 6= Episode closed, final cost of care payment not yet sent. |

| | | | | | | |
|------------|--------------|-------|--|---|---|--|
| KIDS_TRANS | id_prsn | FLOAT | | N | Y | |
| KIDS_TRANS | id_case | FLOAT | | N | Y | |
| KIDS_TRANS | id_epsd | FLOAT | | N | Y | |
| KIDS_TRANS | cd_role | FLOAT | | Y | | |
| KIDS_TRANS | cd_kids | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger10 | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger20 | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger60 | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger61 | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger62 | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger65 | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger70 | FLOAT | | Y | | |
| KIDS_TRANS | fl_trigger79 | FLOAT | | Y | | |
| KIDS_TRANS | ts_cr | DATE | | Y | | |
| KIDS_TRANS | ts_up | DATE | | Y | | |
| KIDS_TRANS | id_cr | FLOAT | | Y | | |
| KIDS_TRANS | id_up | FLOAT | | Y | | |

New Online Processing:

1. Save processing on Out of Home Placement page:

New logic must be added to the Out of Home Placement page (SM10a).
 When ever a new placement is created, then EPISODE.cd_kids = 1.

2. Save trigger processing:

New trigger logic must be added to the topics:

- SM13b – Assets and Employment
- CM02 – Worker Assignment
- CM01 – Person Management
- PM08 – Provider Address

Whenever an update is made to the following pages a trigger should saved to

ONLINE_TRIGGER table.

| Page Name | Trigger Table values |
|---|---|
| Asset/Employment - Assets/Liabilities Tab. (i.e. a save to the ASSET table) | cd_txn = 100 (This value needs to be added to CODE_DESC.id_grp = ONLNTRGR) id_cr = current user ts_cr = current time id_case = current case on page id_prsn = current person on page id_trigger = next sequential number ts_efct_frm = current time |
| Asset/Employment – Employment Tab. (i.e. a save to the EMPLOYMENT table) | cd_txn = 101 (This value needs to be added to CODE_DESC.id_grp = ONLNTRGR) id_cr = current user ts_cr = current time id_case = current case on page id_prsn = current person on page id_trigger = next sequential number ts_efct_frm = current time |
| Create Worker Assignment | cd_txn = 102 (This value needs to be added to CODE_DESC.id_grp = ONLNTRGR) id_cr = current user ts_cr = current time id_case = current case on page id_prsn = current person on page id_trigger = next sequential number ts_efct_frm = current time |

| Page Name | Trigger Table values |
|---|--|
| Person Management-Basic Tab (Any field but Marital Status) and Special Needs page | cd_txn = 103 (This value needs to be added to CODE_DESC.id_grp = ONLNTRGR) id_cr = current user ts_cr = current time id_case = current case on page id_prsn = current person on page id_trigger2 = KIDS_TRANS.id_epsd ts_efct_frm = current time |
| Removed | Cd_txn = 104 |
| Person Management (Address Tab) | cd_txn = 105 (This value needs to be added to CODE_DESC.id_grp = ONLNTRGR) id_cr = current user ts_cr = current time id_case = current case on page id_prsn = current person on page id_trigger2 = KIDS_TRANS.id_epsd ts_efct_frm = current time |
| Maintain Physical Address | cd_txn = 106 (This value needs to be added to CODE_DESC.id_grp = ONLNTRGR) id_cr = current user ts_cr = current time id_trigger = the id_prvd_org from the Address Maintance page ts_efct_frm = current time |

Selection Logic: Run and complete in the order of logic A, logic B, and finally logic C.

A. Update Transmission

I. Read online triggers

- 1) Lookup to the ONLINE_TRIGGER table and retrieve all records cd_txn >= 100 and =< 105
- 2) If cd_txn = 100, then use ONLINE_TRIGGER.id_case and ONLINE_TRIGGER.id_prsn to lookup to KIDS_TRANS where

- KIDS_TRANS.cd_kids = 2 and
- populate WS-CHILD with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = C
 - populate WS-PARENT1 with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = M
 - populate WS_PARENT1 with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = D
 - Then go to Process Description and perform Header + Trigger65
- 3) If cd_txn = 101, then use ONLINE_TRIGGER.id_case and ONLINE_TRIGGER.id_prsn to lookup to KIDS_TRANS where KIDS_TRANS.cd_kids = 2 and
- populate WS-CHILD with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = C
 - populate WS-PARENT1 with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = M
 - populate WS_PARENT1 with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = D
 - Then go to Process Description and perform Header + Trigger62
- 4) If cd_txn = 102, then use ONLINE_TRIGGER.id_case and lookup to EPISODE.id_case and retrieve all rows where EPISODE.cd_kids = 2.
- Then go to Process Description and perform Header + Trigger18
 - Loop through every row retrieved above and create a header and Trigger18 for each row.
- 5) If cd_txn = 103, then use ONLINE_TRIGGER.id_prsn and lookup to KIDS_TRANS.id_prsn where KIDS_TRANS.cd_kids = 2.
- If KIDS_TRANS.cd_role = C, then populate WS-CHILD with KIDS_TRANS.id_prsn
 - If KIDS_TRANS.cd_role = M, then populate WS-PARENT1 with KIDS_TRANS.id_prsn
 - populate WS_PARENT1 with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = D
 - then go to Process Description and perform Header + Trigger60. (NOTE: a single trigger60 will be created for every ONLINE_TRIGGER.id_prsn)
- 6) If cd_txn = 104, Removed
- 7) If cd_txn = 105, then use ONLINE_TRIGGER.id_prsn and lookup

to KIDS_TRANS.id_prsn where KIDS_TRANS.cd_kids = 2.

- If KIDS_TRANS.cd_role = C, then populate WS-CHILD with KIDS_TRANS.id_prsn
 - If KIDS_TRANS.cd_role = M, then populate WS-PARENT1 with KIDS_TRANS.id_prsn
 - populate WS_PARENT1 with KIDS_TRANS.id_prsn where KIDS_TRANS.cd_role = D
 - then go to Process Description and perform Header + Trigger70. (NOTE: a single trigger70 will be created for every ONLINE_TRIGGER.id_prsn)
- 8) If cd_txn = 106, then use ONLINE_TRIGGER.id_trigger2 and perform lookup to EPISODE.id_prvd_org and select those rows with EPISODE.cd_kids = 2.
- For records found perform Header + Trigger79.
- 9) After header and trigger are created, delete row on ONLINE_TRIGGER and loop until no rows selected on ONLINE_TRIGGER cd_txn >= 100 and =< 106

B. Close Episode Transmission

- I. Read Episode table and retrieve episodes where EPISODE.dt_end <> null and EPISODE.cd_kids = 2.
- For records found perform:
Header(with EPISODE.id_prsn) + Trigger20
 - When header/trigger record created update EPISODE.cd_kids = 6.
 - Also update for all rows with this id_epsd on
KIDS_TRANS.cd_kids = 3

C. New Episode Transmission

- I. Read Episode table for new episode
- Sort all EPISODE records by dt_bgn
 - The batch should select those episode records for
 - EPISODE.cd_kids = 1 AND
 - EPISODE.dt_end = null
 - AND
 - EPISODE.fl_cs_court_ordr = Y
 - OR EPISODE.fl_best_intrst = Y and EPISODE.fl_long_term = Y.
 - AND If the selected record has EPISODE.cd_sw_rpt_grp = (67,68,69) (ie Unlicensed Relative Care), then do NOT select.
 - AND Use the EPISODE.cd_srvc to lookup to SERVICE_TYPE, and if the SERVICE_TYPE.fl_pmnt_allwd = Y, then select the

record.

- AND Use EPISODE.id_epsd to look up APPROVAL.id_wrk_type and APPROVAL.cd_actn = 'A' and APPROVAL.cd_stat = 'A' and APPROVAL.cd_wrk_type = 13. If the record exists, select the record.

-

II. Find the Parents

- (Loop through step III for every episode select above and go through the process description and return to III again) In order to send a record to KIDS we must have at least one parent recorded. The Process Description must be performed for episode participant, mother and father if both of the parents are known. Before processing move null to WS-PARENT1 and WS-PARENT2.
 - (Step 1- PERSON lookup) If using EPISODE.id_case and EPISODE.id_prsn to lookup to PERSON and check if id_mom or id_dad is populated. If id_mom not null, store id_mom in WS-ID-PARENT1. If id_dad not null, store id_dad in WS-ID-PARENT2. If id_dad and id_mom = null, then proceed to Step 2. If id_dad = null and id_mom = not null, then proceed to Step 3. If id_mom = null and id_dad = not null, then proceed to Step 4.
 - (Step 2- CASE_PART lookup) Using EPISODE.id_case and EPISODE.id_prsn lookup to CASE_PART and check if cd_rltm_type = 13 or 8 (i.e., a direct child). If true, then retrieve CASE_PART.id_prsn for CASE_PART.id_case = EPISODE.id_case and CASE_PART.cd_rltm_type = 21 (Reference Person). Store CASE_PART.id_prsn in WS-PARENT1.
 - (Step 3- CASE_PART lookup) Using EPISODE.id_case and EPISODE.id_prsn lookup to CASE_PART and check if cd_rltm_type = 13 or 8 (i.e., a direct child). If true, then retrieve CASE_PART.id_prsn for CASE_PART.id_case = EPISODE.id_case and CASE_PART.cd_rltm_type = 21 (Reference Person). If CASE_PART.id_prsn = id_mom, then continue, else store id_prsn in WS-PARENT2.
 - (Step 4- CASE_PART lookup) Using EPISODE.id_case and EPISODE.id_prsn lookup to CASE_PART and check if cd_rltm_type = 13 or 8 (i.e., a direct child). If true, then retrieve CASE_PART.id_prsn for CASE_PART.id_case = EPISODE.id_case and CASE_PART.cd_rltm_type = 21 (Reference Person). If CASE_PART.id_prsn = id_dad, then continue, else store id_prsn in WS-PARENT1.
 - (Step 5) Produce an exception report that lists all EPISODE.id_epsd and EPISODE.id_prsn that had a least WS-PARENT2 or WS-PARENT1 not populated.

- There will either be two or three loops for step III.
 - For the first pass set WS-COUNTER = 1 and use EPISODE.id_prsn.
 - For the second pass set WS-COUNTER = 2 and use WS-PARENT1. If WS-PARENT1 = null skip to third pass.
 - For the third pass set WS-COUNT = 3 and use WS-PARENT2. If WS-PARENT2 = null skip third pass.

III. Update EPISODE table

- When done with this episode change EPISODE.cd_kids = 2 (sent to KIDS).
- Also Update KIDS_TRANS with three rows. One row with the Child, One row with Parent1, and another row with Parent2
 - KIDS_TRANS.id_prsn = EPISODE.id_prsn or WS-PARENT1 or WS-PARENT2
 - KIDS_TRANS.id_case = EPISODE.id_case
 - KIDS_TRANS.id_epsd = EPISODE.id_epsd
 - KIDS_TRANS.cd_role = C (if row is for child), = M (if row is for Parent1), or = D (if row is for parent2)
 - KIDS_TRANS.cd_kids = 2
 - KIDS_TRANS.fl_trigger10 = Y, if trigger 10 is sent
 - KIDS_TRANS.fl_trigger20 = Y, if trigger 20 is sent
 - KIDS_TRANS.fl_trigger60 = Y, if trigger 60 is sent
 - KIDS_TRANS.fl_trigger61 = Y, if trigger 61 is sent
 - KIDS_TRANS.fl_trigger62 = Y, if trigger 62 is sent
 - KIDS_TRANS.fl_trigger65 = Y, if trigger 65 is sent
 - KIDS_TRANS.fl_trigger70 = Y, if trigger 70 is sent
 - KIDS_TRANS.fl_trigger79 = Y, if trigger 79 is sent
 - KIDS_TRANS.ts_cr = current date
 - KIDS_TRANS.ts_up = current date
 - KIDS_TRANS.id_cr = batch id
 - KIDS_TRANS.id_up = batch id

Dependencies: Calculate Ongoing Batch (b-fm01-calc-ongoing.cbl)

Input Files: cycle-date-override

Output Files: Flat File (TOKIDSDAILYMMDDCCYY)

| | | |
|-------------------------|----------------|------|
| Database Tables: | EPISODE | R, U |
| | ONLINE_TRIGGER | R, D |
| | PERSON | R |
| | ADDRESS | R |
| | CASE | R |
| | CODE_DESC | R |
| | EMPLOYMENT | R |
| | ASSET | R |
| | ELIGIBILITY | R |
| | ELIG_REDETR | |

Process Description:

Create a flat file with the following layout. Each record consists of a header (61 characters) plus trigger (232 characters). An update transmission (Logic A) will have the following information:

Header(person ONLINE_TRIGGER.id_prsn) + Trigger(indicated in Step I).

An update transmission with a close episode (Logic B) will have the following information:

Header(with EPISODE.id_prsn) + Trigger20

The format will be the following for a single new episode (Logic C):

Loop 1 (WS-COUNT = 1)

Header(person 00) + Trigger10 (sent just once)

Header(person 00) + Trigger18 (sent just once)

Header(person 00) + Trigger60

Header(person 00) + Trigger61

Header(person 00) + Trigger65

Header(person 00) + Trigger70 (possible multiple)

Header(person 00) + Trigger79
 Loop 2 (WS-COUNT = 2)
 Header(person 01) + Trigger60
 Header(person 01) + Trigger61
 Header(person 01) + Trigger62 (possible multiple)
 Header(person 01) + Trigger65
 Header(person 01) + Trigger70 (possible multiple)
 Header(person 01) + Trigger79
 Loop 3 (WS-COUNT = 3)
 Header(person 02) + Trigger60
 Header(person 02) + Trigger61
 Header(person 02) + Trigger62 (possible multiple)
 Header(person 02) + Trigger65
 Header(person 02) + Trigger70 (possible multiple)
 Header(person 02) + Trigger79

*Note: All alphanumeric values will be capitalized.

| Position | Length | Field Desc | post to layout |
|----------|--------|--|--|
| Header | | | |
| 1 | 14 | WiSACWIS internal Case ID | EPISODE.id_case |
| 2 | 8 | WiSACWIS internal case | EPISODE.id_epsd |
| 3 | 2 | Trigger value. Possible values include 10 = open/change (placement trigger) 20 = close (placement trigger) 12, 17, 60, 62, 65, 70, 18, 79, 78 | Put the number of the succeeding trigger. If EPISODE is closed then = 20 (Logic B) if case is open then = 10 (Logic C) |
| 4 | 8 | Date the flat file was created | Current date or cycle date override (CCYYMMDD) |
| 5 | 10 | WiSACWIS Participant ID | EPISODE.id_prsn |

| Position | Length | Field Desc | post to layout |
|----------|--------|--|--|
| 6 | 3 | FIPS code | Use EPISODE.id_case and perform lookup to EPISODE.cd_cnty. Select tx_desc_lrg (pos. 3-5) From code_desc where id_grp = "CNTYFIPS" and id_grpi = EPISODE.cd_cnty |
| 7 | 10 | WiSACWIS primary worker ID | Using EPISODE.id_case lookup to ASSIGNMENT.id_grp_lvl1 where cd_asgn_ctgry = 1 and cd_asgn_role = 1 and retrieve ASSIGNMENT.id_prsn and fl_open = Y |
| 8 | 2 | Number Order Sequence. | Number each trigger created for this episode. |
| 9 | 2 | 00 = child, 01 = parent one, 02 = parent two | If WS-COUNT =1, then field = '00' If WS-COUNT = 2, then field = '01' If WS-COUNT = 3, then field = '02' If Trigger79 is generated, then field = '03' |
| 10 | 2 | Placement Type | If cd_sw_rpt_grp of SERVICE_TYPE = 63 then post '01' (Kinship Longterm) If cd_sw_rpt_grp = |

| Position | Length | Field Desc | post to layout |
|---|--------|------------|---|
| | | | 74 (Kinship – CHIPS) or 75 (Kinship Non-CHIPS) or 61 (Kinship – CHPS Provider) or 64 (Kinship care no court order) or 432 (Kinship County Funded) or 431 (Kinship Waitlisted) then post ‘02’ (Kinship Short term) If cd_srvc <> 74, 61, 64, 432, 431, 63 then post ‘03’ (Foster Care) |
| Trigger 10/20 (Record 21 + Filler Length 211 = 232) | | | |
| 1 | 1 | FFP Code | If the above Position 3 = 20, then send null, If the above Position 10, = 01 or 02, then display ‘P’ ELSE Using EPISODE.id_case and EPISODE.id_prsn, perform lookup to ELIG_REDET for the most recent dt_cmplt. If no record found perform lookup to ELIGIBILITY for the most recent dt_cmplt. Evaluate ELIG_REDET.cd_stat (or ELIGIBILITY.cd_stat). IF cd_stat = 1 post “R” to field |

| Position | Length | Field Desc | post to layout |
|--|--------|----------------------------------|---|
| | | | IF cd_stat = 4 post "P" to field IF cd_stat = 2 or 3, post "N" to field |
| 2 | 4 | WiSACWIS end reason code. | If this is a close placement then post EPISODE.cd_end_rsn |
| 3 | 8 | Placement Start Date | EPISODE.dt_bgn (CCYYMMDD) |
| 4 | 8 | Placement End Date | If this is a close placement then post EPISODE.dt_trns_end (CCYYMMDD) |
| 5 | 211 | filler | spaces |
| Trigger 12 Good Cause (Record Length 39 + Filler 193 = 232) (not used by WiSACWIS. This will be incorporated into WiSACWIS for Statewide rollout) | | | |
| 1 | 10 | Non-Custodial Parent ID | Not used in WiSACWIS |
| 2 | 10 | Kinship Caregiver's WiSACWIS Pin | Not used in WiSACWIS |
| 3 | 8 | Good Cause File Date | Not used in WiSACWIS |
| 4 | 4 | Good Cause Reason | Not used in WiSACWIS <hr/> APND = pending adoption ASLG = adopt counsel ECHD = EMOT harm child EPRT = EMOT harm parent IEVD = insufficient evidence IRAP = incest-rape NGRT = not granted PCHD = physical harm child PPRT = physical harm parent |

| Position | Length | Field Desc | post to layout |
|---|--------|--|---|
| 5 | 1 | Good Cause Proceed Indicator | Not used in WiSACWIS |
| 6 | 8 | Good Cause End | Not used in WiSACWIS |
| 7 | 191 | filler | |
| Trigger 17 (Record Length 96 + Filler Length 136 = 232) (not used by WiSACWIS) | | | |
| 1 | 10 | Parent's WiSACWIS Pin | not used |
| 2 | 12 | Court docket number | not used |
| 3 | 1 | Method of Payment | not used |
| 4 | 2 | Type of support: Child Support (CS) or Family Support (FS) | not used |
| 5 | 10 | Child support amount (PIC S9(08)V99) | not used |
| 6 | 2 | Child support frequency | not used |
| 7 | 10 | Child support last payment amount (PIC S9(08)V99) | not used |
| 8 | 8 | Child support last payment date | not used |
| 9 | 4 | Child support payment due code | not used |
| 10 | 5 | Child support percentage (PIC 9(03)V99) | not used |
| 11 | 10 | Child support arrearage amount (PIC 9(08)V99) | not used |
| 12 | 1 | Direct payment indicator | not used |
| 13 | 3 | Order county address | not used |
| 14 | 16 | Court city name | not used |
| 15 | 2 | Court state name | not used |
| 16 | 136 | Filler | not used |
| Trigger 18 (Record Length 11 + Filler Length 221 = 232) | | | |
| 1 | 10 | worker id | Using EPISODE.id_case perform lookup to ASSIGNMENT.id_grp_lvl1 for cd_asgn_ctgry = 1 and cd_asgn_role = 1 and |

| Position | Length | Field Desc | post to layout |
|---|--------|---------------------------|--|
| | | | fl_open = Y and dt_end is null retrieve ASSIGNMENT.id_prsn |
| 2 | 1 | Send 'Y' | Send 'Y' |
| 3 | 221 | filler | spaces |
| Trigger 60 (Record Length 94 + Filler Length 138 = 232) If WS-COUNT = 1, use EPISODE.id_prsn If WS-COUNT = 2, use WS-PARENT1 If WS-COUNT = 3, use WS-PARENT2 | | | |
| 1 | 10 | Participant ID | If WS-COUNT = 1, use EPISODE.id_prsn If WS-COUNT = 2, use WS-PARENT1 If WS-COUNT = 3, use WS-PARENT2 |
| 2 | 15 | Participant First Name | retrieve PERSON.nm_frst |
| 3 | 15 | Participant Middle Name | retrieve PERSON.nm_mdl_intl |
| 4 | 20 | Participant Last Name | retrieve PERSON.nm_lst |
| 5 | 3 | Participant Suffix | retrieve PERSON.nm_sfx |
| 6 | 8 | Participant date of birth | retrieve PERSON.dt_brth (CCYYMMDD) |
| 7 | 1 | Participant Gender | retrieve PERSON.cd_gndr |
| 8 | 4 | Participant ethnicity | retrieve PERSON.cd_hspnc = Y, then post '08' (Hispanic) If cd_hpsnc <> Y, then retrieve PERSON.cd_race and post value to flat file. |

| Position | Length | Field Desc | post to layout |
|----------|--------|---|--|
| | | | cd_race = 1 = White cd_race = 2 = Black/African American cd_race = 4 = Asian cd_race = 5 = American Indian/Alaskan Native cd_race = 6 = Unable to Determine cd_race = 7 = Native Hawaiian/Other Pacific Islander |
| 9 | 9 | Participant social security | retrieve PERSON.cd_ssn |
| 10N | 2 | Disability Indicator | Perform lookup with id_prsn to SPECIAL_NEEDS and retrieve cd_prim_basis <hr/> 10 = Developmental Delay (CFS-40) 7 = Medically Fragile 3 = Member of a Minority Group 2 = Member of a Sibling Group 6 = Mental Disability &/or Learn. Disability 9 = Other 5 = Physical Disability 4 = Serious Emotional Maladjustment 8 = Visually/Hearing Impaired |
| 11N | 1 | Teen Parent's Child receives kinship (i.e. Teen Parent's Kinship Provider gets two payments. 1) for Teen Parent | use WS-PARENT1 (ie mother) and retrieve PERSON.fl_rcv_knshp_py |

| Position | Length | Field Desc | post to layout |
|--|--------|-----------------------------|--|
| | | 2) for Teen Parent's Child) | mnt |
| 12N | 1 | Child is receiving MA | lookup to MEDICAID_ELIGIBILITY .id_prsn and retrieve fl_ma where ts_up (max) |
| 13N | 1 | Marital Status field | If WS-COUNT=1, then post 7 to flat file If WS-COUNT = 2, then lookup to PERSON.cd_mthrs_crnt_sts If WS-COUNT = 3, then lookup to PERSON.cd_fthrs_crnt_sts Perform the translation and post to file: 1 change to 2 (married) 2 change to 3 (divorced) 3 change to 5 (separated) 7 change to 6 (Other) 8 change to 7 (Unknown) 4 change 1 (Never Married) 6 change to 4 (widowed) <u>5 change to 9 (Deceased)</u> |
| 14N | 4 | Paternity Disposition | <u>Not used in WiSACWIS</u> |
| 15 | 138 | filler | spaces |
| <p>Trigger 62 (Record Length 155 + Filler Length 77 = 232) (Do not post this record for Child) If WS-COUNT = 2, use WS-PARENT1 If WS-COUNT = 3, use WS-PARENT2</p> <p>Loop through and retrieve all EMPLOYMENT.id_prsn records for each person.</p> | | | |

| Position | Length | Field Desc | post to layout |
|----------|--------|--------------------|--|
| 1 | 10 | Participant ID | <p>If WS-COUNT = 1, use EPISODE.id_prsn</p> <p>If WS-COUNT = 2, use WS-PARENT1</p> <p>If WS-COUNT = 3, use WS-PARENT2</p> |
| 2 | 31 | employer name | <p>use the appropriate person id above and EPISODE.id_case and retrieve EMPLOYMENT.tx_employr</p> |
| 3 | 31 | employer address 1 | <p>use the appropriate person id above and EPISODE.id_case and lookup to EMPLOYMENT and use id_empl to retrieve (and concatenate)</p> <p>ADDRESS.ad_strt_nbr and ad_strt_nme and ad_apt for ADDRESS.cd_grp = 8 and ADDRESS.cd_grp_type = 1</p> |
| 4 | 31 | employer address 2 | <p>use the appropriate person id above and EPISODE.id_case and lookup to EMPLOYMENT and use id_empl to retrieve ADDRESS.ad_ln2 for ADDRESS.cd_grp = 8 and ADDRESS.cd_grp_type = 1</p> |
| 4 | 16 | employer city | <p>use the appropriate person id above and EPISODE.id_case and lookup to EMPLOYMENT and use id_empl to retrieve ADDRESS.tx_town for</p> |

| Position | Length | Field Desc | post to layout |
|---|--------|------------------|---|
| | | | ADDRESS.cd_grp = 8 and ADDRESS.cd_grp_type = 1 |
| 5 | 2 | employer state | use the appropriate person id above and EPISODE.id_case and lookup to EMPLOYMENT and use id_empl to retrieve ADDRESS.cd_state for ADDRESS.cd_grp = 8 and ADDRESS.cd_grp_type = 1 |
| 6 | 9 | employer zip | use the appropriate person id above and EPISODE.id_case and lookup to EMPLOYMENT and use id_empl to retrieve ADDRESS.cd_zip for ADDRESS.cd_grp = 8 and ADDRESS.cd_grp_type = 1 |
| 7 | 25 | employer country | use the appropriate person id above and EPISODE.id_case and lookup to EMPLOYMENT and use id_empl to retrieve ADDRESS.cd_cntry for ADDRESS.cd_grp = 8 and ADDRESS.cd_grp_type = 1 Using cd_cntry perform lookup to CODE_DESC.id_grp = 'COUNTRY' and post to flat file tx_desc_lrg |
| 8 | 77 | filler | spaces |
| Trigger 65 (Record Length 96 + Filler Length 136 = 232) | | | |

| Position | Length | Field Desc | post to layout |
|--|--------|-------------------------|--|
| If WS-COUNT = 1, use EPISODE.id_prsn If WS-COUNT = 2, use WS-PARENT1 If WS-COUNT = 3, use WS-PARENT2 | | | |
| 1 | 10 | Participant ID | If WS-COUNT = 1, use EPISODE.id_prsn If WS-COUNT = 2, use WS-PARENT1 If WS-COUNT = 3, use WS-PARENT2 |
| 2 | 31 | insurance carrier name | use the appropriate person id above and EPISODE.id_case and retrieve ASSET.tx_desc1 for ASSET.cd_aset_type = 25 and ASSET.cd_ins_type = 6 or 9 |
| 3 | 15 | insurance policy number | use the appropriate person id above and EPISODE.id_case and retrieve ASSET.tx_desc4 for ASSET.cd_aset_type = 25 and ASSET.cd_ins_type = 6 or 9 |
| 4N | 30 | Name of Policy Holder | use the appropriate person id above and EPISODE.id_case and retrieve ASSET.tx_desc3 for ASSET.cd_aset_type = 25 and ASSET.cd_ins_type = 6 or 9 |
| 5N | 10 | Medicaid Number | Not used in WiSACWIS |
| 6 | 136 | filler | spaces |

| Position | Length | Field Desc | post to layout |
|---|--------|------------------|---|
| Trigger 70 (Record Length 118 + Filler Length 114 = 232) If WS-COUNT = 1, use EPISODE.id_prsn If WS-COUNT = 2, use WS-PARENT1 If WS-COUNT = 3, use WS-PARENT2 Loop through and retrieve all ADDRESS.cd_grp_type(s) for each person. | | | |
| 1 | 10 | Participant ID | If WS-COUNT = 1, use EPISODE.id_prsn If WS-COUNT = 2, use WS-PARENT1 If WS-COUNT = 3, use WS-PARENT2 |
| 2 | 5 | apartment number | use the appropriate person id above and lookup to ADDRESS.id_adrss_grp and retrieve ADDRESS.ad_aprt for ADDRESS.cd_grp = 3 and ADDRESS.dt_efct_end = Null |

| Position | Length | Field Desc | post to layout |
|----------|--------|----------------|---|
| 3 | 31 | Address line 1 | <p>use the appropriate person id above and</p> <p>lookup to ADDRESS.id_adrss_grp and retrieve</p> <p>ADDRESS.ad_strt_nbr and ad_strt_nme for</p> <p>ADDRESS.cd_grp = 3</p> <p>and</p> <p>ADDRESS.dt_efct_end = Null</p> |
| 4 | 31 | Address line 2 | <p>use the appropriate person id above and</p> <p>lookup to ADDRESS.id_adrss_grp and retrieve</p> <p>ADDRESS.ad_ln3 for</p> <p>ADDRESS.cd_grp = 3</p> <p>and</p> <p>ADDRESS.dt_efct_end = Null</p> |
| 5 | 16 | city | <p>use the appropriate person id above and</p> <p>lookup to ADDRESS.id_adrss_grp and retrieve</p> <p>ADDRESS.tx_town for</p> <p>ADDRESS.cd_grp = 3</p> <p>and</p> <p>ADDRESS.dt_efct_end = Null</p> |

| Position | Length | Field Desc | post to layout |
|----------|--------|------------|--|
| 6 | 2 | State | use the appropriate person id above and lookup to ADDRESS.id_adrss_grp and retrieve ADDRESS.cd_state for ADDRESS.cd_grp = 3 and ADDRESS.dt_efct_end = Null |
| 7 | 9 | Zip | use the appropriate person id above and lookup to ADDRESS.id_adrss_grp and retrieve ADDRESS.cd_zip for ADDRESS.cd_grp = 3 and ADDRESS.dt_efct_end = Null |
| 8 | 10 | telephone | use the appropriate person id above and lookup to ADDRESS.id_adrss_grp and retrieve ADDRESS.ad_phone for ADDRESS.cd_grp = 3 and ADDRESS.dt_efct_end = Null |

| Position | Length | Field Desc | post to layout |
|--|--------|---------------------|---|
| 9 | 4 | Address type code | <p>After the above address has been posted, evaluate cd_grp. If cd_grp = 1, then post 'RESD'</p> <p>if cd_grp = 2, then post "MAIL"</p> <p>If cd_grp = 3, then post 'RES2'</p> <p>If cd_grp = 5, then post 'RSPP'</p> <p>If cd_grp = 6, then post 'LSKN'</p> <p>If cd_grp = 7, then post 'PRNT'</p> <p>If cd_grp = 8, then post 'NPRT'</p> <p>If cd_grp = 9, then post 'WORK'</p> |
| 10 | 114 | filler | |
| <p>Trigger 78 (Record Length 20 + Filler Length 212 = 232)</p> <p>Loop through PERSON_MERG.id_prsn_fmr and select rows where ts_up (date only) = PARMDATE.</p> <p>Take PERSON_MERG.id_prsn_fmr and lookup to KIDS_TRANS.id_prsn where cd_kids = 2 or 3</p> | | | |
| 1N | 10 | Incorrect Person ID | Lookup to PERSON_MERG.id_prsn_fmr and display this. |
| 2N | 10 | Correct Person ID | Lookup to PERSON_MERG.id_prsn_fmr where and retrieve id_prsn_rtn |
| 3N | 212 | Filler | |
| <p>Trigger 79 (Record Length 232 + Filler Length 0 = 232)</p> | | | |

| Position | Length | Field Desc | post to layout |
|---|--------|----------------------------|---|
| Only perform this section if using EPISODE.cd_srvc = lookup to SERVICE_TYPE.cd_srvc where SERVICE_TYPE.cd_sw_rpt_grp in (75,74,61,63,64,432,431) – Kinship Statewide Reporting Group. | | | |
| 1N | 10 | Provider ID | Use EPISODE.id_prvd_org and lookup to PROVIDER_PART.id_prvd_org and retrieve id_prsn where cd_role = 1 and cd_stat = A |
| 2N | 34 | Provider Name Last | For the EPISODE retrieved, use EPISODE.id_prvd_org to lookup to and retrieve PROVIDER_ORG.tx_prvd_nm |
| 3N | 15 | Provider Name First | For the EPISODE retrieved, use EPISODE.id_prvd_org to lookup to and retrieve PROVIDER_ORG.nm_prvd_fst |
| 4N | 30 | Provider Address number | Using id_prvd_org selected above retrieve ADDRESS.ad_strt_nbr for ADDRESS.cd_grp = 5 Insert one space and, Also retrieve ADDRESS.ad_strt_nme for ADDRESS.cd_grp = 5 |
| 5N | 4 | Provider Address apartment | Using id_prvd_org selected above retrieve ADDRESS.ad_apt for ADDRESS.cd_grp = 5 |
| 6N | 36 | Provider Address line 2 | Using id_prvd_org selected above retrieve ADDRESS.ad_ln2 for ADDRESS.cd_grp = 5 |
| 7N | 12 | Provider Address line 3 | Using id_prvd_org selected above retrieve ADDRESS.ad_ln3 for |

| Position | Length | Field Desc | post to layout |
|----------|--------|----------------------------------|--|
| | | | ADDRESS.cd_grp = 5 |
| 8N | 2 | Provider Address State | Using id_prvd_org selected above retrieve ADDRESS.cd_state for ADDRESS.cd_grp = 5 |
| 9N | 9 | Provider Address Zip | Using id_prvd_org selected above retrieve ADDRESS.cd_zip for ADDRESS.cd_grp = 5 |
| 10N | 40 | Provider Address Town | Using id_prvd_org selected above retrieve ADDRESS.tx_town for ADDRESS.cd_grp = 5 |
| 11N | 10 | Provider Address Phone | Using id_prvd_org selected above retrieve ADDRESS.ad_phone for ADDRESS.cd_grp = 5 |
| 12N | 4 | Provider Address Phone Extension | Using id_prvd_org selected above retrieve ADDRESS.ad_ext for ADDRESS.cd_grp = 5 |
| 13N | 1 | Provider Parent 1 Gender | Using id_prvd_org selected above perform lookup to PROVIDER_PART and retrieve id_prsn for cd_role = 1. If no record found post 'U' to flat file. If record found lookup to PERSON.id_prsn and retrieve cd_gndr, and post value to flat file. |
| 14N | 9 | Provider Parent 1 SSN | Using id_prvd_org selected above perform lookup to PROVIDER_PART and retrieve id_prsn for cd_role = 1. If no record found post 'U' to flat file. If record found lookup to PERSON.id_prsn and |

| Position | Length | Field Desc | post to layout |
|----------|--------|-----------------------|--|
| | | | retrieve cd_ssn, and post value to flat file. |
| 15N | 8 | Date of Birth | Using id_prvd_org selected above perform lookup to PROVIDER_PART and retrieve id_prsn for cd_role = 1. If record found retrieve PERSON.dt_brth |
| 16N | 2 | Ref Person | Not used in WiSACWIS |
| 17N | 2 | Relationship to Child | Not used in WiSACWIS |
| 18N | 4 | Ethnicity | <p>Using id_prvd_org selected above perform lookup to PROVIDER_PART and retrieve id_prsn for cd_role = 1. If record found retrieve PERSON.cd_hspnc = Y, then post '08' (Hispanic) If cd_hpsnc <> Y, then retrieve PERSON.cd_race and post value to flat file.</p> <hr/> <p>cd_race = 1 = White cd_race = 2 = Black/African American cd_race = 4 = Asian cd_race = 5 = American Indian/Alaskan Native cd_race = 6 = Unable to Determine cd_race = 7 = Native Hawaiian/Other Pacific Islander</p> |

1.4.5 WiSACWIS to KIDS Monthly Batch

Program Name: b-fm0707-kids-financial

Script Name:

Process Summary: WiSACWIS will transmit on a monthly basis financial cost of care information to KIDS. The batch will give to KIDS ongoing cost of care amounts. Additionally, two kinds of rescinded payments will be sent. The first rescinded payment type will be cancelled payments. The second rescinded payment type will be overpayments that have been sent to providers. The KIDS system will apply the balance of these amounts to foster care unreimbursed assistance subaccount for a child

Frequency: Monthly

Dependencies: 1. DOA to WiSACWIS Checks Printed (b-fm02-chk-printed.cbl)

Input Files: cycle-date-override
fm07-kids-financial-parm (with from-date and to-date)

Output Files: flat file (KIDSMONTHLYMMDDCCYY)

Database Tables: EPISODE R
PAYMENT R

Selection Logic: **Step I**

Using cycle-date-override perform lookup to EPISODE and retrieve those rows where EPISODE.cd_kids = 2 or EPISODE.cd_kids = 6.

- 1) For each record found use EPISODE.id_epsd to perform lookup to PAYMENT.id_epsd and retrieve positive payment records where from-date =< CHECKS.dt_disp =< to-date and PAYMENT.cd_pmnt_type = G or C or T and dt_chck <> null. When a row is found proceed to Process Description.

- 2) Also use EPISODE.id_case and EPISODE.id_prsn and select PAYMENT records where from-date =< CHECKS.dt_disp =< to-date and where PAYMENT.dt_rqst >= EPISODE.dt_bgn and where (PAYMENT.cd_pmnt_type = O and PAYMENT.cd_srvc = 10011 (ie. one time placement payment)) and dt_chck <> null. When a row is found proceed to Process Description.
- 3) Also retrieve negative payments (-). Using EPISODE.id_prsn and EPISODE.id_case perform lookup to PAYMENT.id_epsd and retrieve records where PAYMENT.am_rqst < 0 and from-date =< CHECKS.dt_disp =< to-date and store PAYMENT.am_rqst to WS-RESCIND-AMT. When a row is found proceed to Process Description.
- 4) Also retrieved cancelled payment (C). Perform lookup to PAYMENT.id_epsd and retrieve records where PAYMENT.cd_pmnt_stat = C and use id_chck to lookup to CHECKS where from-date =< CHECKS.dt_disp =< to-date and CHECKS.cd_chck_disp = S or E or C or D. Add PAYMENT.am_rqst to WS-RESCIND-AMT. When a row is found proceed to Process Description.

When record is found proceed with Process Description, then return to select next PAYMENT row with next id_epsd. For id_epsd which no PAYMENT row is found, issue an exception report.

When finished, evaluate records retrieved where EPISODE.cd_kids = 6.

For each id_epsd verify no other payment rows exist where PAYMENT.cd_pmnt_stat = null and PAYMENT.id_chck is null

For rows where above is true, set EPISODE.cd_kids = 3.

To prevent episode and payment records from getting stuck in the holding stage due to unforeseen workflow issues, perform separate lookup:

Evaluate all database records where EPISODE.cd_kids = 6. If

EPISODE.dt_end precedes current date minus 90 days set

EPISODE.cd_kids = 3.

Step II

When completed with Step I above, loop through EPISODE and find row where cd_kids = 3. Using EPISODE.id_prsn make sure all EPISODE rows for this id_prsn have dt_end <> null. If true, use id_prsn and lookup to PAYMENT and sum all am_nontrust for this id_prsn. If sum = 0, then create a record as described in the Process Description below and set pos24 = Y. Excluded population of pos13 – pos23. When this record is added for this id_prsn then perform update using id_epsd setting EPISODE.cd_kids = 4 and KIDS_TRANS.cd_kids = 4.

Process Description:

For each episode found create a flat file with the following retrieval logic.

| Position | Field Length | Description | Where to retrieve in WiSACWIS |
|----------|--------------|----------------------------------|---|
| 1 | 14 | Case ID | EPISODE.id_case |
| 2 | 10 | Episode ID | EPISODE.id_prsn |
| 3 | 3 | Rep Unit | Filler |
| 4 | 19 | Last Name | Use EPISODE.id_prsn and EPISODE.id_case and retrieve PERSON.nm_last |
| 5 | 10 | First Name | Use EPISODE.id_prsn and EPISODE.id_case and retrieve PERSON.nm_frst |
| 6 | 1 | Mid Int | Use EPISODE.id_prsn and EPISODE.id_case and retrieve PERSON.nm_mdl_intl |
| 7 | 3 | Name Suffix | Use EPISODE.id_prsn and EPISODE.id_case and retrieve PERSON.nm_sfx |
| 8 | 2 | Run Date Century | Post cycle-date-override century |
| 9 | 2 | Run Date Year | Post cycle-date-override year |
| 10 | 2 | Run Date Month | Post cycle-date-override month |
| 12 | 2 | Run Date Day | Post cycle-date-override day |
| 13 | 2 | Transaction Date CC | PAYMENT.dt_chck (pos. 5,6) |
| 14 | 2 | Transaction Date YY | PAYMENT.dt_chck (pos. 7,8) |
| 15 | 2 | Transaction Date MM | PAYMENT.dt_chck (pos. 1,2) |
| 16 | 2 | Transaction Date DD | PAYMENT.dt_chck (pos. 3,4) |
| 17 | 2 | Transaction Code (KIDS internal) | post '01' |

| | | | |
|-----|----|-----------------------|--|
| | | use) | |
| 18 | 8 | Payment Amount | If coming from Step 1 or 2 set = PAYMENT.am_rqst If coming from Step 3 or 4 set = WS-RESCIND-AMT and set value to negative |
| 19N | 8 | Payment Begin Date | PAYMENT.dt_pmnt_begin (CCYYMMDD) |
| 20N | 8 | Payment End Date | PAYMENT.dt_pmnt_end (CCYYMMDD) |
| 21N | 15 | Payment ID | If coming from Step 1,2 or 4, PAYMENT.id_pmnt If coming from Step 3, PAYMENT_ADJUST.ovp_id_pmnt |
| 22N | 1 | Payment not received | If coming from Step 3, set = C. If coming from Step 4, set = Y else set = spaces |
| 23N | 2 | Kinship Payment | Using PAYMENT.cd_plcm_setng and lookup to CODE_DESC.id_grp = PLCMSTNG and if CODE_DESC.cd_cdesc_type = 4, then post 01, else post 02 to flat file. |
| 24N | 1 | Cost of Care Recouped | Value is Y or null depending on selection logic above. |
| 25 | 5 | FIPS code | Post FIPS Code (CODE_DESC.tx_desc_lrg) using CODE_DESC.id_grp = 'CNTYFIPS' and id_grpi = PAYMENT.cd_cnty_new |

1.4.6 EFT Ledger Update batch

Program Name: b-fm0702-eft-ledger-update

Script Name: b-fm07-00.script

Process Summary: This batch will read the daily file from Firststar and update any deposits or withdrawals to the Trust Account Ledger. The batch takes the claim number from the EFT file and performs a lookup to the Benefit table. When a match is found, then the money is deposited in the ledger for the child with this trust account.

There are several messages that are printed on the exception report. If duplicate claim numbers are found in the Benefit table this will be stated in the exception report as: "Duplicate Claim Number." When a claim is number is not found, this is also listed on the exception report as: "Claim Number Not Found." When negative transaction is made, the report will display: "Negative Transaction." When a positive transaction is successfully made, the report will display: "Deposit Posted to Ledger." When a duplicate bank account number is found, this is listed on the exception report as: "Duplicate Bank Account Number". When there is no bank account number found, this is listed on the exception report as: "Bank Account Number not Found." If there are duplicate trust account numbers for the person, the exception is listed as: "Duplicate Trust Account ID" and if no trust account is found, the exception is listed as: "Trust Account ID not found".

There are seven deposit transactions types that are made by this batch. When a deposit is made it will look to see if any Personal Needs deduction should be made and the batch will create a separate deposit entry for called Personal Needs. This benefit transaction will have a new amount that is the original EFT amount minus the personal needs amount. The batch will processes only transactions that are generated by SSI-State, SSI-Federal, SSA-Survivor's, SSA-Disability, VA and Railroad benefits. SSA does not distinguish on the EFT file a difference between SSA Survivor's and SSA Disability. The assumption is made that a child would only in very rare instances receive both benefits concurrently. When money is received from SSA, the batch will lookup to the Benefit table find the benefit type the child is receiving using the claim number. If

that claim number is used for both SSA-Survivor's and SSA-Disability, then no ledger posting will occur, and the exception report will display this transaction as a "Duplicate Claim Number."

Frequency: Daily.

Dependencies: None

Input Files: cycle-date-override

Output Files: Exception Report (FM07-eft-excpt.rpt)

| | | |
|-------------------------|----------------------|---------|
| Database Tables: | TRUST_ACCOUNT | R |
| | BENEFIT | R |
| | TRUST_ACCOUNT_LEDGER | C, R, U |
| | BANK | R |

Process Description:

Record 1 - File Header Record – identifies the origin and destination of the entries contained in the file. This record also indicates that all records contain 94 characters and that each block contains 10 records. The file header record also contains the transmission date to further identify the file.

Record 5 – Company/Batch Header Record – identifies the batch, sender and receiver. Files may have on batch or more than one batch, depending upon company needs. This record indicates whether the money is received from SSI-State, SSA-Survivor's, SSA-Disability, VA, Railroads, or SSI-Federal.

Record 6 – Entry Detail Record – provides record of each transaction. Contains all information necessary to process the debit, credit or prenote as described by the transaction code. Each child will have a unique row with deposit information.

Record 7-Addendum Record- provides additional information that is in free form text. Each agency used this section differently. This section is not used by SSI-State.

Record 8 – Company/Batch Control Record – immediately follows the last Detail Record (6 Record). The function of this record is to provide totals for all entries in a batch. The batch number is the same as the on the Batch Header Record (Record 5).

Record 9 – File Control Record – provides totals of entire file for control purposes. This is the last record before the trailer records or any labels.

The records in the file layout received from Firststar will be as follows:

Record 1

Record 5

Record 6 (multiples indicated in 5 Record)

Record 7 (multiples following each Record 6)

Record 8

Record 5

Record 6 (multiples indicated in 5 Record)

Record 7 (multiples following each Record 6)

Record 8

Record 9

1. Retrieve the first 5 Record. This will have '5' in the first position.
2. Read first 5 Record and store position 41-50 in WS-FED-TAX-ID
 - 2.1. if position 41-50 value = 1751093604 , then the coming records are cd_bnft_typ = 2 (SSI-State)
 - 2.2. if position 54-64 value = "SUPP SEC" , then the coming records are cd_bnft_typ = 6 (SSI-Federal)
 - 2.3. if position 54-64 value = "SOC SEC" then the coming records are cd_bnft_typ = 1 (SSA-Disability) or cd_bnft_typ = 7 (SSA-Survivor's)
 - 2.4. if position 54-64 value = "RR" , then the coming records are cd_bnft_typ = 8 (Railroads Retirees) See Step 3.3 sublogic.
 - 2.4. 2.5. if position 54-64 value = "VA" , then the coming records are cd_bnft_typ = 4 (VA)
3. There are two methods to extract child identification and deposit amount.
 - 3.1. If cd_bnft_typ = 2, then retrieve the first following 6 Record (this will have a '6' in the first position). Take the Bank Account Number (Record6, pos. 13-29) and perform lookup to BANK.id_bank. If row is found, then use id_bank and lookup up TRUST_ACCOUNT.id_trust_acct. Use TRUST_ACCOUNT.id_trust_acct to lookup BENEFIT.id_trust_acct. If row is found, use id_trust_acct to lookup BENEFIT.id_clm and BENEFIT.cd_bnft_type. Match BENEFIT.id_clm to SSN (Record6, pos. 40-54). If

row is found, then use BENEFIT.id_trust_acct and lookup TRUST_ACCOUNT_LEDGER. Store Records6, (pos. 55-76) in WS-NAME.

3.1.1. Exception Report processing:

- 3.1.1.1. If more than one row returned, issue exception report with this transaction for this WS-NAME and SSN and display “Duplicate Claim Number in System.”
- 3.1.1.2. If no row found on BENEFIT table, issue exception report for this WS-NAME and SSN and display message “Claim number not found in benefit table.”
- 3.1.1.3. If row is found, and TRUST_ACCOUNT.dt_closing <> null, then issue message “Closed Account” and to NOT post this amount to ledger.
- 3.1.1.4. If more than one than one row returned with same bank account numbers, issue exception report with this transaction for this WS-NAME and SSN, display message “Duplicate Bank Account Number.”
- 3.1.1.5. If no row found on BANK table, issue exception report for this WS-NAME and SSN and display message “Bank Account Number Not Found.”
- 3.1.1.6. If more than one row returned with the same trust account number, issue exception report with this transaction for this WS-NAME and SSN, display message “Duplicate Trust Account Number.”
- 3.1.1.7. If no row found on the TRUST ACCOUNT table, issue exception report for this WS-NAME and SSN and display message “Trust Account Not Found.”
- 3.1.1.8. If no row found on BENEFIT table, issue exception report for this WS-NAME and SSN and display message “Benefit Type Not Found.”

3.2. If cd_bnft_typ = 6, 8 or 4, then retrieve the first following 6 Record and 7 Record pair (this will have a ‘6’/’7’ in the first position). Take the Bank Account Number (Record6, pos. 13-29) and perform lookup to BANK.id_bank. If row is found, then use id_bank and lookup up TRUST_ACCOUNT.id_trust_acct. Use TRUST_ACCOUNT.id_trust_acct to lookup BENEFIT.id_trust_acct. If row is found, use id_trust_acct to lookup BENEFIT.id_clm and BENEFIT.cd_bnft_type. Match BENEFIT.id_clm to SSN (Record6, pos. 40-54). If row is found, then use BENEFIT.id_trust_acct and lookup to TRUST_ACCOUNT_LEDGER. Store Record7, (pos. 39-64) in WS-NAME.

3.2.1. Exception Report processing:

- 3.2.1.1. If more than one row returned, issue exception report with this transaction for this WS-NAME and SSN and display “Duplicate Claim Number in System.”

- 3.2.1.2. If no row found on BENEFIT table, issue exception report for this WS-NAME and SSN and display message “Claim number not found in benefit table.”
 - 3.2.1.3. If row is found, and TRUST_ACCOUNT.dt_closing <> null, then issue message “Closed Account” and to NOT post this amount to ledger.
 - 3.2.1.4. If more than one row returned with same bank account numbers, issue exception report with this transaction for this WS-NAME and SSN, display message “Duplicate Bank Account Number.”
 - 3.2.1.5. If no row found on BANK table, issue exception report for this WS-NAME and SSN and display message “Bank Account Number Not Found.”
 - 3.2.1.6. If more than one row returned with the same trust account number, issue exception report with this transaction for this WS-NAME and SSN, display message “Duplicate Trust Account Number.”
 - 3.2.1.7. If no row found on the TRUST ACCOUNT table, issue exception report for this WS-NAME and SSN and display message “Trust Account Not Found.”
 - 3.2.1.8. If no row found on BENEFIT table, issue exception report for this WS-NAME and SSN and display message “Benefit Type Not Found.”
- 3.3. If cd_bnft_type = 1 or 7, then retrieve the first following 6 Record and 7 Record pair (this will have a ‘6’/’7’ in the first position). Take the Bank Account Number (Record6, pos. 13-29) and perform lookup to BANK.id_bank. If row is found, then use id_bank and lookup up TRUST_ACCOUNT.id_trust_acct. Use TRUST_ACCOUNT.id_trust_acct to lookup BENEFIT.id_trust_acct. If row is found, use id_trust_acct to lookup BENEFIT.id_clm and BENEFIT.cd_bnft_type. Match BENEFIT.id_clm to SSN (Record6, pos. 40-54). If only one row is found, then use BENEFIT.id_trust_acct and lookup to TRUST_ACCOUNT_LEDGER. Store Record7, (pos. 39-64) in WS-NAME.
- 3.3.1. Exception Report processing:
 - 3.3.1.1. If more than one row returned, issue exception report with this transaction for this WS-NAME and SSN and display “Duplicate Claim Number in System.”
 - 3.3.1.2. If no row found on BENEFIT table, issue exception report for this WS-NAME and SSN and display message “Claim number not found in benefit table.”
 - 3.3.1.3. If row is found, and TRUST_ACCOUNT.dt_closing <> null, then issue message “Closed Account” and to NOT post this amount to ledger.

- 3.3.1.4. If more than one than one row returned with same bank account numbers, issue exception report with this transaction for this WS-NAME and SSN, display message “Duplicate Bank Account Number.”
 - 3.3.1.5. If no row found on BANK table, issue exception report for this WS-NAME and SSN and display message “Bank Account Number Not Found.”
 - 3.3.1.6. If more than one row returned with the same trust account number, issue exception report with this transaction for this WS-NAME and SSN, display message “Duplicate Trust Account Number.”
 - 3.3.1.7. If no row found on the TRUST ACCOUNT table, issue exception report for this WS-NAME and SSN and display message “Trust Account Not Found.”
 - 3.3.1.8. If no row found on BENEFIT table, issue exception report for this WS-NAME and SSN and display message “Benefit Type Not Found.”
4. If BENEFIT.am_prsn_nds > 0, set FL-PRSN-NDS = Y
 5. Insert a TRUST_ACCOUNT_LEDGER record with the following attributes:
 - 5.1. TRUST_ACCOUNT_LEDGER.id_case = BENEFIT.id_case
 - 5.2. TRUST_ACCOUNT_LEDGER.id_prsn = BENEFIT.id_prsn
 - 5.3. TRUST_ACCOUNT_LEDGER.id_ldgr = system generated unique ledger ID.
 - 5.4. TRUST_ACCOUNT_LEDGER.dt_trans = Record 5, pos. 70-75. (YYMMDD). Prefix YY with the CC found in cycle-date-override.
 - 5.5. TRUST_ACCOUNT_LEDGER.cd_trans_typ =
 - If Record 6, pos. 02-03 = 22, 32,23, 33 then cd_trans_typ = “1” (i.e. Deposit)
 - If Record 6, pos. 02-03 = 27, 37, 28, 38, then cd_trans_typ = “2” (i.e., Withdrawal). If a withdrawal is made, then also include this in an exception report, for the SSN and WS-NAME and display the message “Negative Amount.”
 - 5.6. TRUST_ACCOUNT_LEDGER.am_trans = If cd_trans_typ = 1, then Record 6, pos 30-39 minus BENEFIT.am_prsnl_nds. If cd_trans_type = 2, then multiply Record 6, pos. 30-39 by negative one.
 - 5.7. TRUST_ACCOUNT_LEDGER.cd_detl = If cd_trans_typ = 1, then cd_detl = 7 (i.e. EFT Deposit). If cd_trans_typ = 2, then cd_detl = 10 (i.e., EFT Withdrawal)
 - 5.8. TRUST_ACCOUNT_LEDGER.cd_bnft_typ = the cd_bnft_typ of the one determined in step (2)
 - 5.9. TRUST_ACCOUNT_LEDGER.tx_desc = Null
 - 5.10. TRUST_ACCOUNT_LEDGER.id_cr = Batch ID
 - 5.11. TRUST_ACCOUNT_LEDGER.ts_cr = current system date

- 5.12. TRUST_ACCOUNT_LEDGER.id_up = Batch ID
- 5.13. TRUST_ACCOUNT_LEDGER.ts_up = current system date
- 5.14. TRUST_ACCOUNT_LEDGER.id_trust_acct = the id_trust_acct of BENEFIT record selected
- 5.15. TRUST_ACCOUNT_LEDGER.id_pmnt = Null
- 5.16. TRUST_ACCOUNT_LEDGER.am_rem =
 - If cd_trans_typ = 1, am_rem = am_trans
 - If cd_trans_typ = 2, am_rem = 0
6. If WS-PRSNL-NDS = Y, then insert a second TRUST_ACCOUNT_LEDGER record (else continue to Step (8)) with the following attributes:
 - 6.1. TRUST_ACCOUNT_LEDGER.id_case = BENEFIT.id_case
 - 6.2. TRUST_ACCOUNT_LEDGER.id_prsn = BENEFIT.id_prsn
 - 6.3. TRUST_ACCOUNT_LEDGER.id_ldgr = system generated unique ledger ID.
 - 6.4. TRUST_ACCOUNT_LEDGER.dt_trans = Record 5, pos. 70-75. (YYMMDD). Prefix YY with the CC found in cycle-date-override.
 - TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1 (i.e. Deposit)
 - 6.5. TRUST_ACCOUNT_LEDGER.am_trans = BENEFIT.am_prsnl_nds
 - 6.6. TRUST_ACCOUNT_LEDGER.cd_detl = 7 (i.e. EFT Deposit)
 - 6.7. TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 3 (i.e Personal Needs)
 - 6.8. TRUST_ACCOUNT_LEDGER.tx_desc = Null
 - 6.9. TRUST_ACCOUNT_LEDGER.id_cr = Batch ID
 - 6.10. TRUST_ACCOUNT_LEDGER.ts_cr = current system date
 - 6.11. TRUST_ACCOUNT_LEDGER.id_up = Batch ID
 - 6.12. TRUST_ACCOUNT_LEDGER.ts_up = current system date
 - 6.13. TRUST_ACCOUNT_LEDGER.id_trust_acct = the id_trust_acct of BENEFIT record selected
 - 6.14. TRUST_ACCOUNT_LEDGER.id_pmnt = Null
 - 6.15. TRUST_ACCOUNT_LEDGER.am_rem =
 - If cd_trans_typ = 1, am_rem = am_trans
 - If cd_trans_typ = 2, am_rem = 0
7. If insert into TRUST_ACCOUNT_LEDGER is successful, then issue message: "Deposit Posted to Ledger."
8. Return to Step (3), and retrieve next Record6.

9. If next record not a Record6, check to see if it is a Record 8. If Record 8, then proceed from Step (2), else end processing.
10. Create Exception Report (FM07-eft-excpt.rpt) as follows

Exception Report

Account Number: 999999999

| Name | Claim Number | Amt | Reason |
|---------|-------------------|--------------|---|
| WS-Name | (Record 6, 40-54) | \$9999999.99 | Negative Amount |
| WS-Name | (Record 6, 40-54) | \$9999999.99 | Claim Number not found in Benefit Table |
| WS-Name | (Record 6, 40-54) | \$9999999.99 | Duplicate Claim Number in system |
| WS-Name | (Record 6, 40-54) | \$9999999.99 | Deposit posted to ledger |
| WS-Name | (Record 6, 40-54) | \$9999999.99 | Deposit posted to ledger (Personal Needs) |
| WS-Name | (Record 6, 40-54) | \$9999999.99 | Closed Account |

1.4.7 KIDS Financial Ledger Update batch

Program Name: b-fm0708-KIDS-ledger-update

Script Name: b-fm07-00.script

Process Summary: This batch will read the monthly file from KIDS and update any deposits or withdrawals to the Trust Account Ledger. This file indicates EFT transactions that have been sent to the WiSACWIS accounts we have specified to KIDS. The batch takes the SACWIS person number (which is stored in KIDS) from the flat file and performs a lookup to the Benefit table. As a child can have multiple trust accounts open, an additional match with county will have to be made to ensure that the money is deposited in the correct account. When a match is found, the money is deposited in the ledger for the child with this trust account.

There are several messages that are printed on the exception report. If duplicate person ID numbers are found in the Benefit table this will be stated in the exception report as: "DUPLICATE ID_PRSN NUMBER IN SYSTEM." This may occur when a general and dedicate account are open, and established to both receive child support for a single child.

When a person ID number is not found on the Trust Account table, the exception report states that “ID_PRSN NOT FOUND ON TRUST ACCOUNT TABLE”. If the county for the benefit from KIDS does not match the county of the child’s trust account, the error states: “KIDS FIPS DOES NOT MATCH COUNTY.” When negative transaction is made, the report will display: “Negative amount - Posted to ledger.” Other error messages displayed include: “NO OPEN TRUST ACCOUNT IN THE COUNTY FOR THE PERSON_ID”, “NO CHILD SUPPORT BENEFIT FOR PERSON_ID”, “NO CHILD SUPPORT BENEFIT IN COUNTY {NAME}”. When a positive transaction is successfully made, the report will display: “Deposit posted to ledger - Regular” or “Deposit posted to ledger - Personal Needs.” When a trust account is closed, a benefit will not be posted and the message will read: “CLOSED ACCOUNT.”

There is a single benefit source that deposit transactions will come from, namely Child Support. When a deposit is made it will look to see if any Personal Needs deduction should be made and the batch will create a separate deposit entry for called Personal Needs. This benefit transaction will have a new amount that is the original EFT amount minus the personal needs amount. Personal Needs is not used in the reimbursement processes, see Topic Paper *FM04a: Reimbursement*.

Frequency: Daily.

Dependencies: None

Input Files: KIDSDEPOSIT

Output Files: Exception Report (FM07-eft-excpt.rpt)
and a verification report.

Database Tables:

| | |
|----------------------|---------|
| TRUST_ACCOUNT | R |
| BENEFIT | R |
| TRUST_ACCOUNT_LEDGER | C, R, U |

Process Description:

File layout for the KIDS input file:

```

01  WS-DISB-EXT.
    05  EXT-DISB-FIPS-ENT.
        10  EXT-DISB-STATE                      PIC X(02).
        10  EXT-DISB-COUNTY-NBR                 PIC X(03).
        10  EXT-DISB-FILL                       PIC X(05).
    05  EXT-DISB-WRKR-RESP                      PIC X(08).
    05  EXT-DISB-ACTIVE-ACCT                   PIC X(10).
    05  EXT-DISB-TAX-TYPE                      PIC X(01).
    05  EXT-DISB-FIPS-INTR-ACCT               PIC X(10).
    05  EXT-DISB-REASON-CODE                  PIC X(01).
    05  EXT-DISB-EVENT-KEY.
        10  EXT-DISB-EVENT-TYPE                PIC X(03).
        10  EXT-DISB-EVENT-BATCH               PIC X(08).
        10  EXT-DISB-EVENT-ITEM                PIC X(05).
    05  EXT-DISB-JNLEVT-RECD.
        10  EXT-DISB-RECDCC                    PIC X(02).
        10  EXT-DISB-RECDYY                    PIC X(02).
        10  EXT-DISB-RECDMM                    PIC X(02).
        10  EXT-DISB-RECDDD                    PIC X(02).
    05  EXT-DISB-PAYEE-NAME.
        10  EXT-DISB-PAYEE-NM-L                PIC X(20).
        10  EXT-DISB-PAYEE-NM-F                PIC X(15).
        10  EXT-DISB-PAYEE-NM-MI              PIC X.
    05  EXT-DISB-ADDTL-NAME.
        10  EXT-DISB-ADDTL-NM-L                PIC X(20).
        10  EXT-DISB-ADDTL-NM-F                PIC X(15).
        10  EXT-DISB-ADDTL-NM-MI              PIC X.
    05  EXT-DISB-PAYEE-ACCT                   PIC X(10).
    05  EXT-DISB-PAYER-ACCT                   PIC X(10).
    05  EXT-DISB-CP-SSN                       PIC X(09).
    05  EXT-DISB-PIN                          PIC X(10).
    05  EXT-DISB-EXT-CASE.
        10  FILLER                            PIC X(02).
        10  SACWIS-PIN                        PIC X(10).
        10  FILLER                            PIC X(02).
    05  EXT-DISB-AM-TRN-APLD                   PIC S9(08)V99.
    05  EXT-DISB-DSB-EVENT                    PIC X(16).
    05  EXT-DISB-ERROR-SW                    PIC X(01).

```

| | | |
|----|---------------------------|------------|
| 05 | EXT-DISB-CHECK-NAME | PIC X(31). |
| 05 | EXT-DISB-CD-DISB-MTH | PIC X(01). |
| 05 | EXT-DISB-CD-COLL-SRC | PIC X(04). |
| 05 | EXT-DISB-CD-COLL-SRC-SORT | PIC X(01). |
| 05 | EXT-DISB-CHK-LESS-THAN-0 | PIC X(01). |
| 05 | EXT-DISB-FILLER | PIC X(46). |

11. Retrieve the following fields

- 11.1. EXT-DISB-ADDTL-NAME
- 11.2. SACWIS-PIN
- 11.3. EXT-DISB-AM-TRN-APLD
- 11.4. EXT-DISB-JNLEVT-RECD
- 11.5. EXT-DISB-PAYEE-ACCT
- 11.6. EXT-DISB-FIPS-INTR-ACCT
- 11.7. EXT-DISB-FIPS-ENT
- 11.8. EXT-DISB-STATE
- 11.9. EXT-DISB-COUNTY-NBR

12. This method should be used to extract child identification and deposit amount.

- 12.1. Retrieve the field SACWIS-PIN of the first row.
- 12.2. Perform lookup to BENEFIT.id_prsn where cd_bnft_type = 5.
- 12.3. Match FIPS county code (EXT-DISB-FIPS-INTR-ACCT pos. 30 – 40) with TRUST_ACCOUNT.cd_cnty for the id_prsn.

12.3.1. Exception Report processing:

- 12.3.1.1. If more than one row returned, issue exception report with this transaction for this EXT-DISB-ADDTL-NAME and SACWIS-PIN and display “DUPLICATE ID_PRSN NUMBER IN SYSTEM.”
- 12.3.1.2. If no row found on BENEFIT table, issue exception report for EXT-DISB-ADDTL-NAME and SACWIS-PIN and display message “ID_PRSN NOT FOUND OR INVALID BENEFIT TYPE.”
- 12.3.1.3. If row is found, and TRUST_ACCOUNT.dt_closing <> null, then issue message “CLOSED ACCOUNT” and do NOT post this amount to ledger.

13. If `BENEFIT.am_prsnl_nds > 0`, set `FL-PRSN-NDS = Y`
14. Insert a `TRUST_ACCOUNT_LEDGER` record with the following attributes:
 - 14.1. `TRUST_ACCOUNT_LEDGER.id_case = BENEFIT.id_case`
 - 14.2. `TRUST_ACCOUNT_LEDGER.id_prsn = BENEFIT.id_prsn`
 - 14.3. `TRUST_ACCOUNT_LEDGER.id_ldgr = system generated unique ledger ID.`
 - 14.4. `TRUST_ACCOUNT_LEDGER.dt_trans = EXT-DISB-JNLEVT-RECD.`
 - 14.5. `TRUST_ACCOUNT_LEDGER.cd_trans_typ =`
 - If `EXT-DISB-AM-TRN-APLD >= 0`, then `cd_trans_typ = "1"` (i.e. Deposit)
 - If `EXT-DISB-AM-TRN-APLD < 0`, then `cd_trans_typ = "2"` (i.e., Withdrawal). If a withdrawal is made, then also include this in an exception report, for `EXT-DISB-ADDTL-NAME` and `SACWIS-PIN` and display the message "Negative Amount."
 - 14.6. `TRUST_ACCOUNT_LEDGER.am_trans =` If `cd_trans_typ = 1`, then `EXT-DISB-AM-TRN-APLD`. If `cd_trans_typ = 2`, then multiply then `EXT-DISB-AM-TRN-APLD` by negative one.
 - 14.7. `TRUST_ACCOUNT_LEDGER.cd_detl =` If `cd_trans_typ = 1`, then `cd_detl = 7` (i.e. EFT Deposit). If `cd_trans_typ = 2`, then `cd_detl = 10` (i.e., EFT Withdrawal)
 - 14.8. `TRUST_ACCOUNT_LEDGER.cd_bnft_typ =` the `cd_bnft_typ` of the one determined in step (2)
 - 14.9. `TRUST_ACCOUNT_LEDGER.tx_desc = Null`
 - 14.10. `TRUST_ACCOUNT_LEDGER.id_cr = Batch ID`
 - 14.11. `TRUST_ACCOUNT_LEDGER.ts_cr = current system date`
 - 14.12. `TRUST_ACCOUNT_LEDGER.id_up = Batch ID`
 - 14.13. `TRUST_ACCOUNT_LEDGER.ts_up = current system date`
 - 14.14. `TRUST_ACCOUNT_LEDGER.id_trust_acct =` the `id_trust_acct` of `BENEFIT` record selected
 - 14.15. `TRUST_ACCOUNT_LEDGER.id_pmnt = Null`
 - 14.16. `TRUST_ACCOUNT_LEDGER.am_rem =`
 - If `cd_trans_typ = 1`, `am_rem = am_trans`
 - If `cd_trans_typ = 2`, `am_rem = 0`
15. If `WS-PRSNL-NDS = Y`, then insert a second `TRUST_ACCOUNT_LEDGER` record (else continue to Step (8)) with the following attributes:
 - 15.1. `TRUST_ACCOUNT_LEDGER.id_case = BENEFIT.id_case`
 - 15.2. `TRUST_ACCOUNT_LEDGER.id_prsn = BENEFIT.id_prsn`

- 15.3. TRUST_ACCOUNT_LEDGER.id_ldgr = system generated unique ledger ID.
- 15.4. TRUST_ACCOUNT_LEDGER.dt_trans = System Date.
- TRUST_ACCOUNT_LEDGER.cd_trans_typ = 1 (i.e. Deposit)
 - 15.5. TRUST_ACCOUNT_LEDGER.am_trans = BENEFIT.am_prsnl_nds
 - 15.6. TRUST_ACCOUNT_LEDGER.cd_detl = 7 (i.e. EFT Deposit)
 - 15.7. TRUST_ACCOUNT_LEDGER.cd_bnft_typ = 3 (i.e Personal Needs)
 - 15.8. TRUST_ACCOUNT_LEDGER.tx_desc = Null
 - 15.9. TRUST_ACCOUNT_LEDGER.id_cr = Batch ID
 - 15.10. TRUST_ACCOUNT_LEDGER.ts_cr = current system date
 - 15.11. TRUST_ACCOUNT_LEDGER.id_up = Batch ID
 - 15.12. TRUST_ACCOUNT_LEDGER.ts_up = current system date
 - 15.13. TRUST_ACCOUNT_LEDGER.id_trust_acct = the id_trust_acct of BENEFIT record selected
 - 15.14. TRUST_ACCOUNT_LEDGER.id_pmnt = Null
 - 15.15. TRUST_ACCOUNT_LEDGER.am_rem =
- If cd_trans_typ = 1, am_rem = am_trans
- If cd_trans_typ = 2, am_rem = 0
- 16. If insert into TRUST_ACCOUNT_LEDGER is successful, then issue message: “Deposit Posted to Ledger.”
- 17. Return to Step (1), and retrieve next record.
- 18. If no next record then create Exception Report (FM07-b-kids-excpt-rpt) and Verification Report (FM07-b-kids-verif-rpt) follows. Only one Exception Report will be generated with each run, and sent to each county for workers to update data in the system. A separate Verification Report will be sent to each county.
 - 18.1. Account Number = BANK.cd_acct_number, where BANK.id_bank = TRUST_ACCOUNT.id_bank and TRUST_ACCOUNT.id_trust_acct = BENEFIT.id_trust_acct for the child in this record processed.
 - 18.2. Create a temporary table with the above children and sort by id_bank so that the report groups children by Account Number. Make separate reports for each BANK.cd_cnty. If no account number if found issue message “CD_ACCT_NUMBER NOT FOUND IN SYSTEM.”

Exception Report

| <u>Name</u> | <u>Claim Number</u> | <u>Amt</u> | <u>Reason</u> |
|---------------------|---------------------|--------------|---|
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | TRUST_ACCOUNT NUMBER NOT FOUND IN SYSTEM |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | DUPLICATE ID_PRSN NUMBER IN SYSTEM |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | ID_PRSN NOT FOUND OR INVALID BENEFIT TYPE |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | CD_ACCT_NUMBER NOT FOUND IN SYSTEM |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | CLOSED ACCOUNT |

Verification Report

Account Number: 999999999 County: XXXXXXXXXX

| <u>Name</u> | <u>Claim Number</u> | <u>Amt</u> | <u>Reason</u> |
|---------------------|---------------------|--------------|---|
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | Deposit posted to ledger - Regular |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | Deposit posted to ledger - Personal Needs |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | Negative amount - Posted to ledger |

Account Number: 888888888 County: XXXXXXXXXX

| <u>Name</u> | <u>Claim Number</u> | <u>Amt</u> | <u>Reason</u> |
|---------------------|---------------------|--------------|---|
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | Deposit posted to ledger - Regular |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | Deposit posted to ledger - Personal Needs |
| EXT-DISB-ADDTL-NAME | SACWIS-PIN | \$9999999.99 | Negative amount - Posted to ledger |

1.4.8 Trust Account Spend Down Tickler Update

Program Name: b-fm0703-spend-down-tickler

Process Summary: This batch will reads the trust account ledger for each child and calculates the current balance for that child's trust account. If the balances is equal to or greater than the maximum balance located on the Maximum Balance page, a tickler will be issued. The tickler reminds the worker to perform a spend down, so that a child does not loose his eligibility status because of excess assets.

Frequency: Daily.

Dependencies: Always run after the following programs have completed: b-fm0701-spread-interest, b-fm0702-eft-ledger-update, b-fm0700-ldgr-dep, b-fm0708-kids-ledger-update.

Input Files: None

Output Files: None

| | | |
|-------------------------|----------------------|------------|
| Database Tables: | TRUST_ACCOUNT | R, U |
| | TICKLER | C, R, U, D |
| | TRUST_ACCOUNT_LEDGER | R |

Process Description:

Sum am_trans for every id_trust_acct, where TRUST_ACCOUNT.dt_closing is null. Next retrieve the row from MAX_BALANCE.

FFP

If the child is FFP, then retrieve the row where
 dt_effectv is max,
 Fl_non_ffp = N,

Issue Tickler

If the sum of am_trans \geq MAX_BALANCE.am_maxbal multiplied by MAX_BALANCE.am_pctng, then issue a tickler if TRUST_ACCOUNT.id_tickler is null.

Delete Tickler

If the sum of am_trans $<$ MAX_BALANCE.am_maxbal multiplied by MAX_BALANCE.am_pctng, then delete tickler if TRUST_ACCOUNT.id_tickler is not null.

Non-FFP

If the child is non-FFP or has no eligiblity rows, then retrieve row from MAX_BALANCE where

Dt_effectv is max,

Fl_non_ffp = Y

Issue Tickler

If the sum of am_trans \geq MAX_BALANCE.am_maxbal multiplied by MAX_BALANCE.am_pctng, then issue a tickler if TRUST_ACCOUNT.id_tickler is null.

Delete Tickler

If the sum of am_trans $<$ MAX_BALANCE.am_maxbal multiplied by MAX_BALANCE.am_pctng, then delete tickler if TRUST_ACCOUNT.id_tickler is not null.

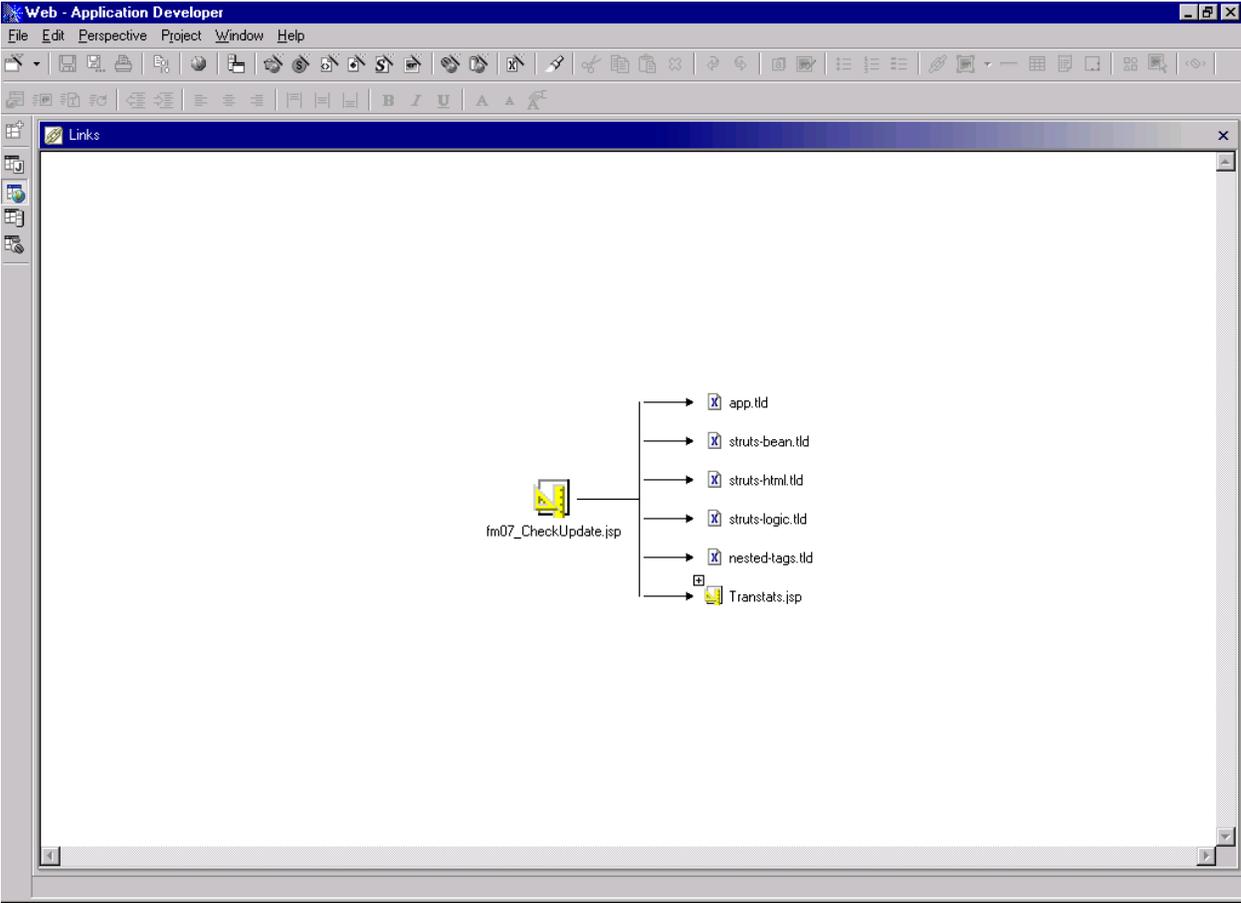
1.4. Online Components

1.4.1. User Interface Components

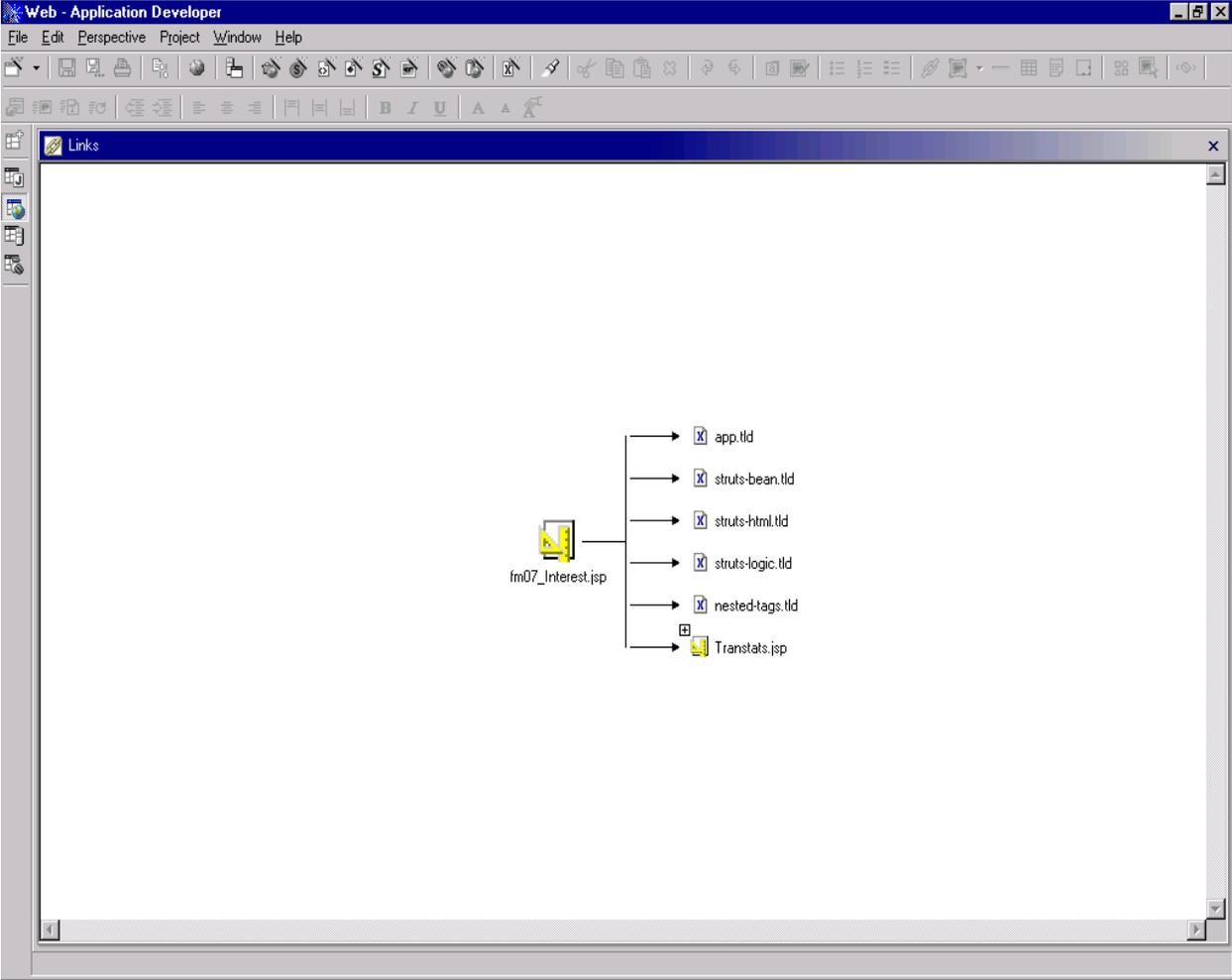
There are three primary JSPs for the CM03 (Checklist) topic, cm03_Checklist.jsp, cm03_ChecklistTemplate.jsp and cm03_SelectChecklist.jsp. Each has a corresponding Javascript file.

Below are the Link diagrams of the Checklist pages.

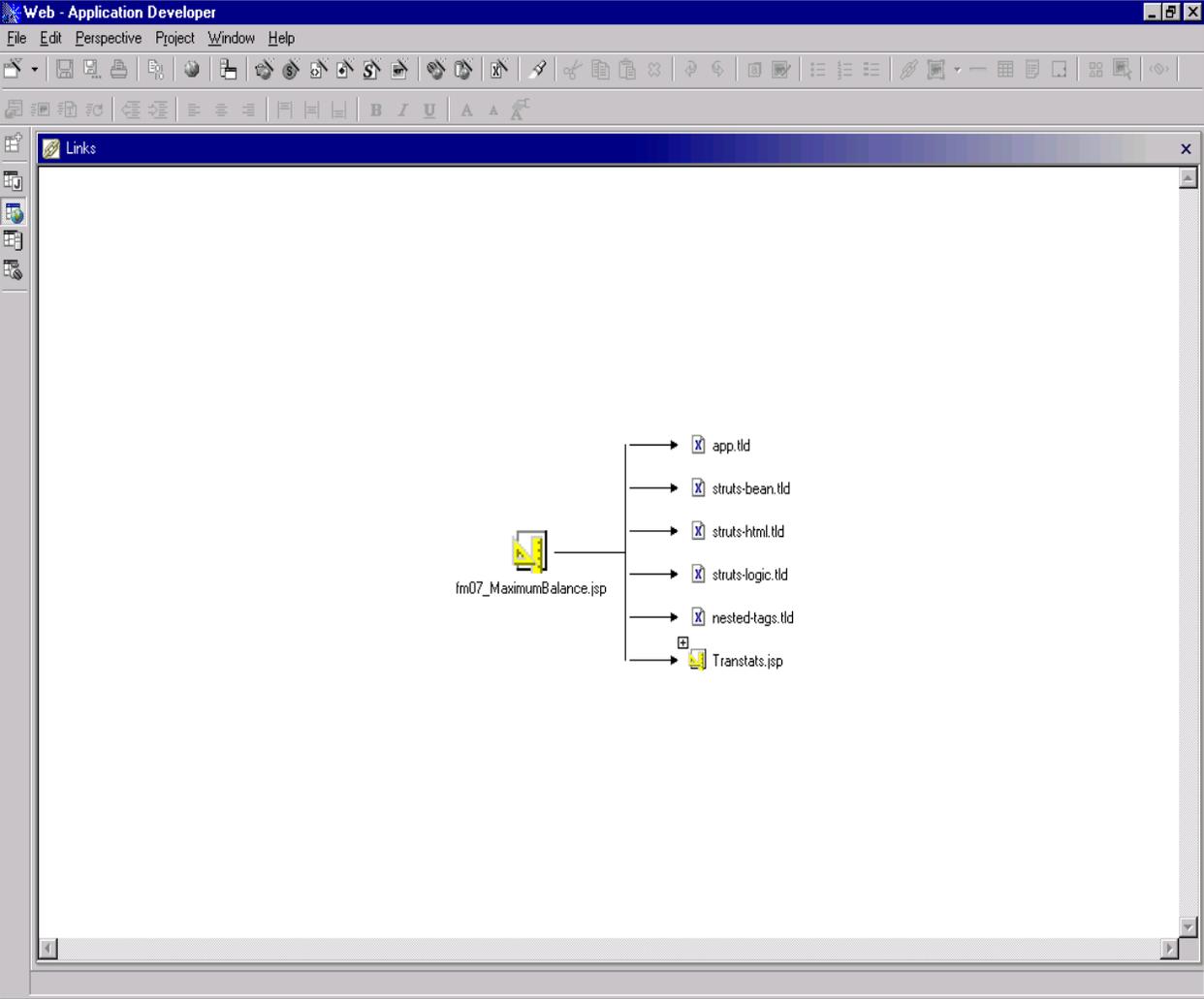
fm07_CheckUpdate.jsp



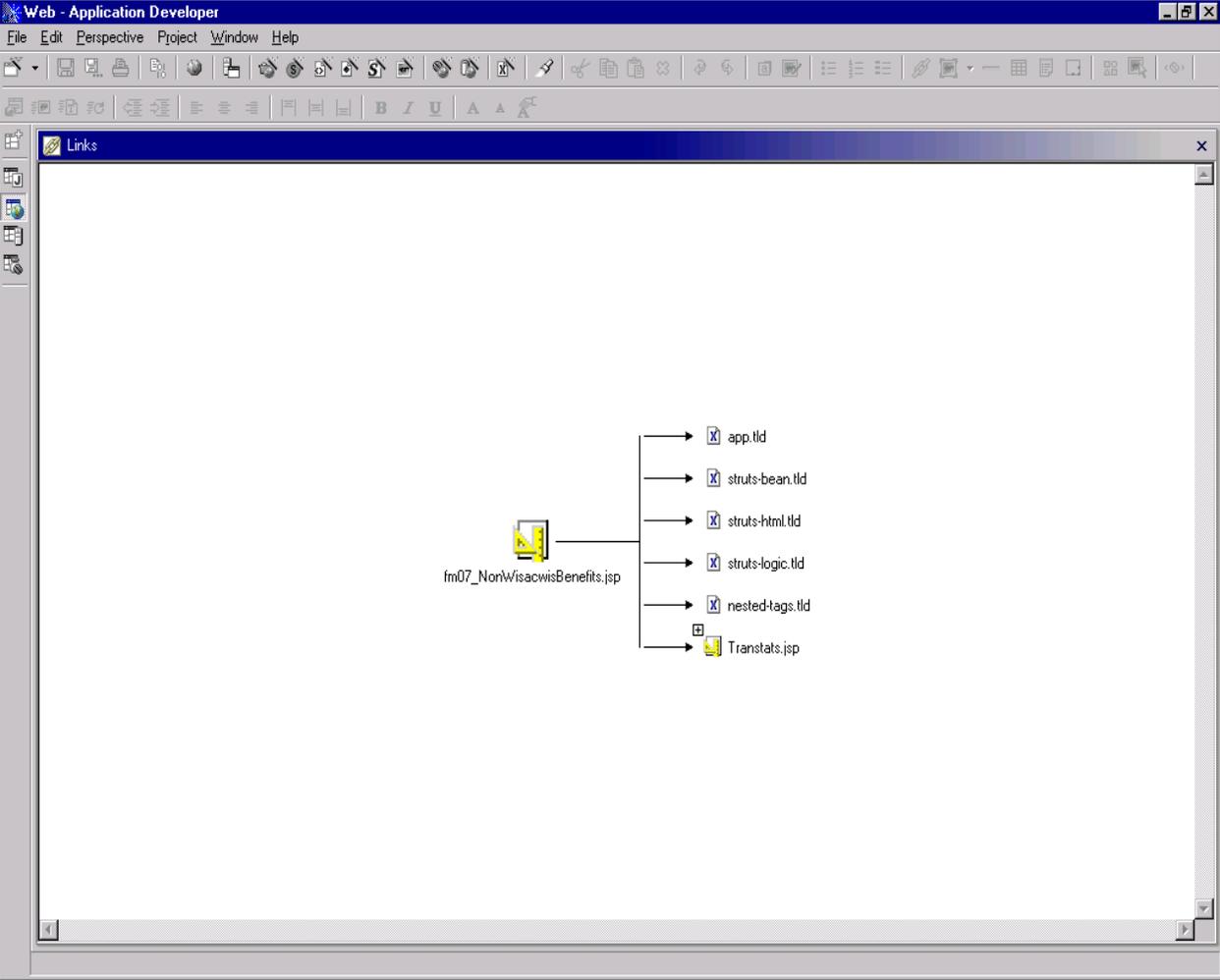
fm07_Interest.jsp



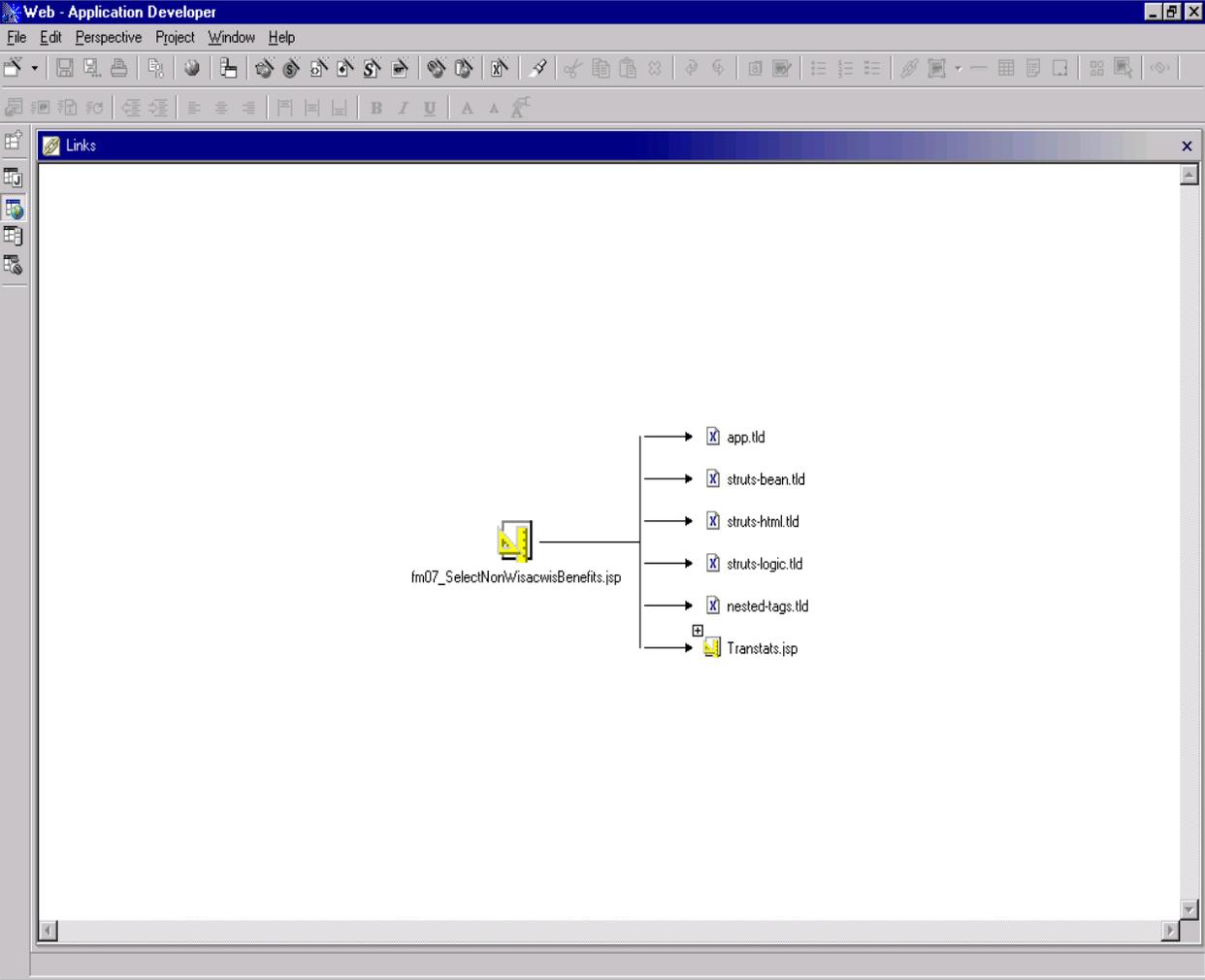
fm07_MaximumBalance.jsp



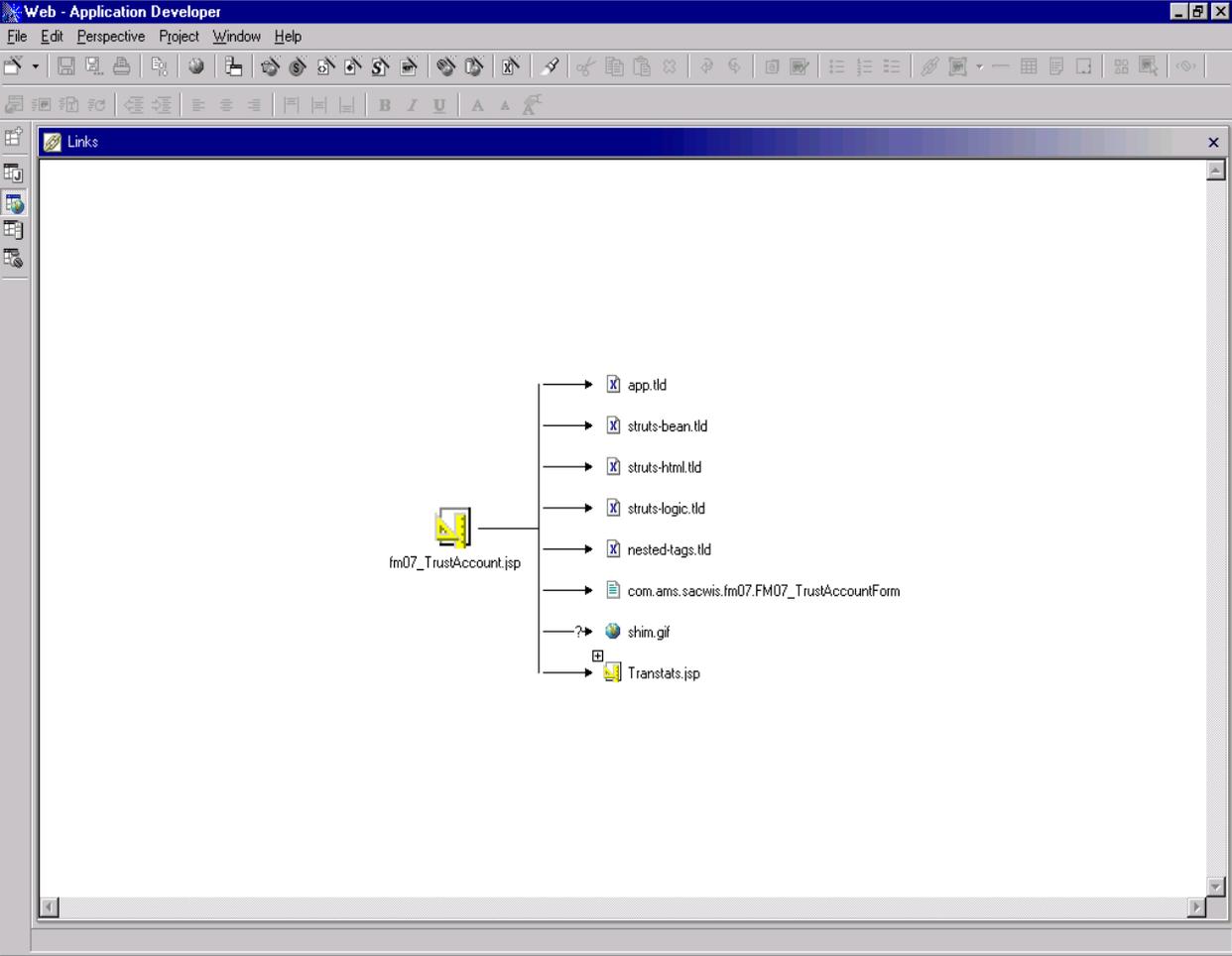
fm07_NonWisacwisBenefits.jsp



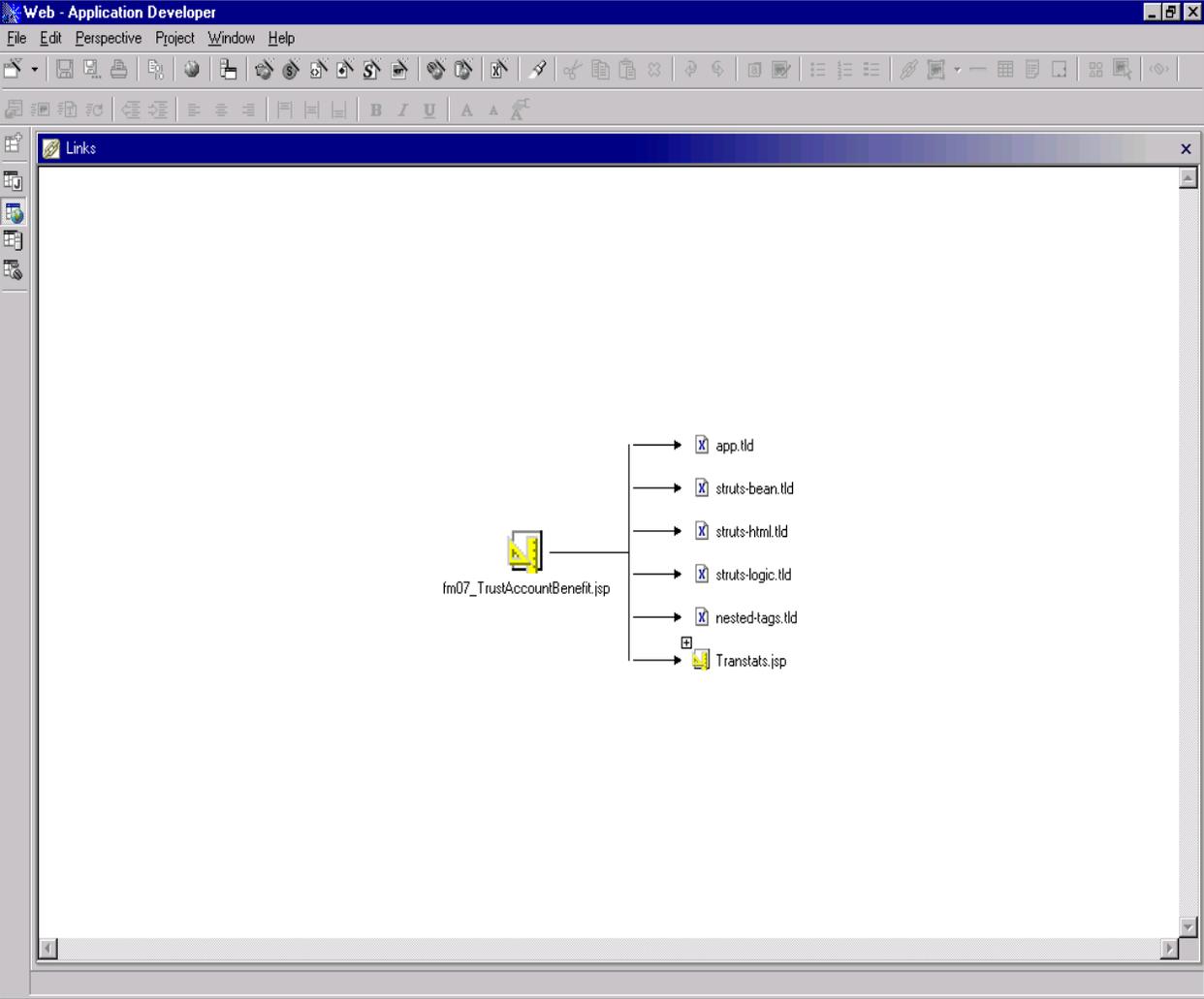
fm07_SelectNonWisacwisBenefits.jsp



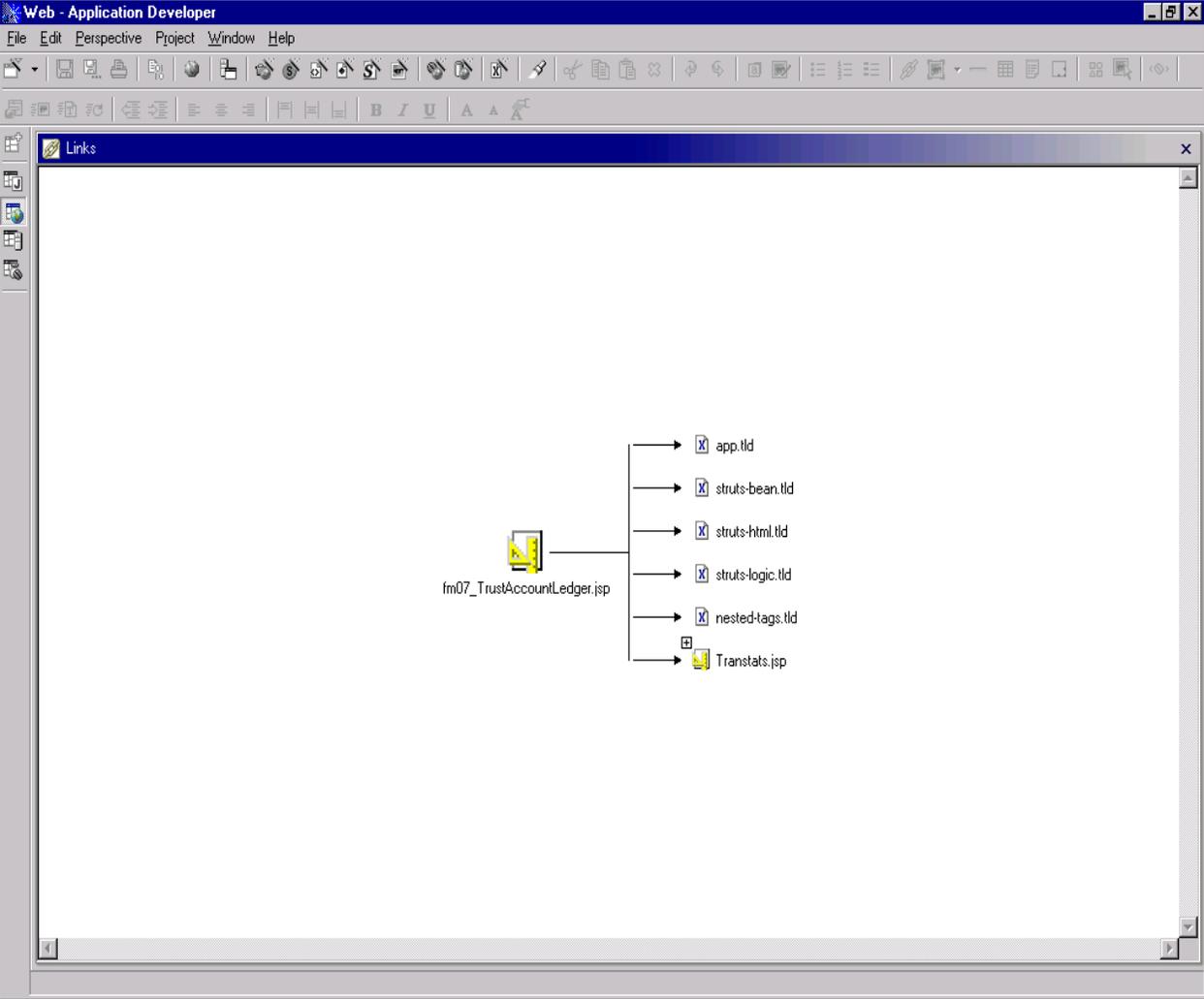
fm07_TrustAccount.jsp



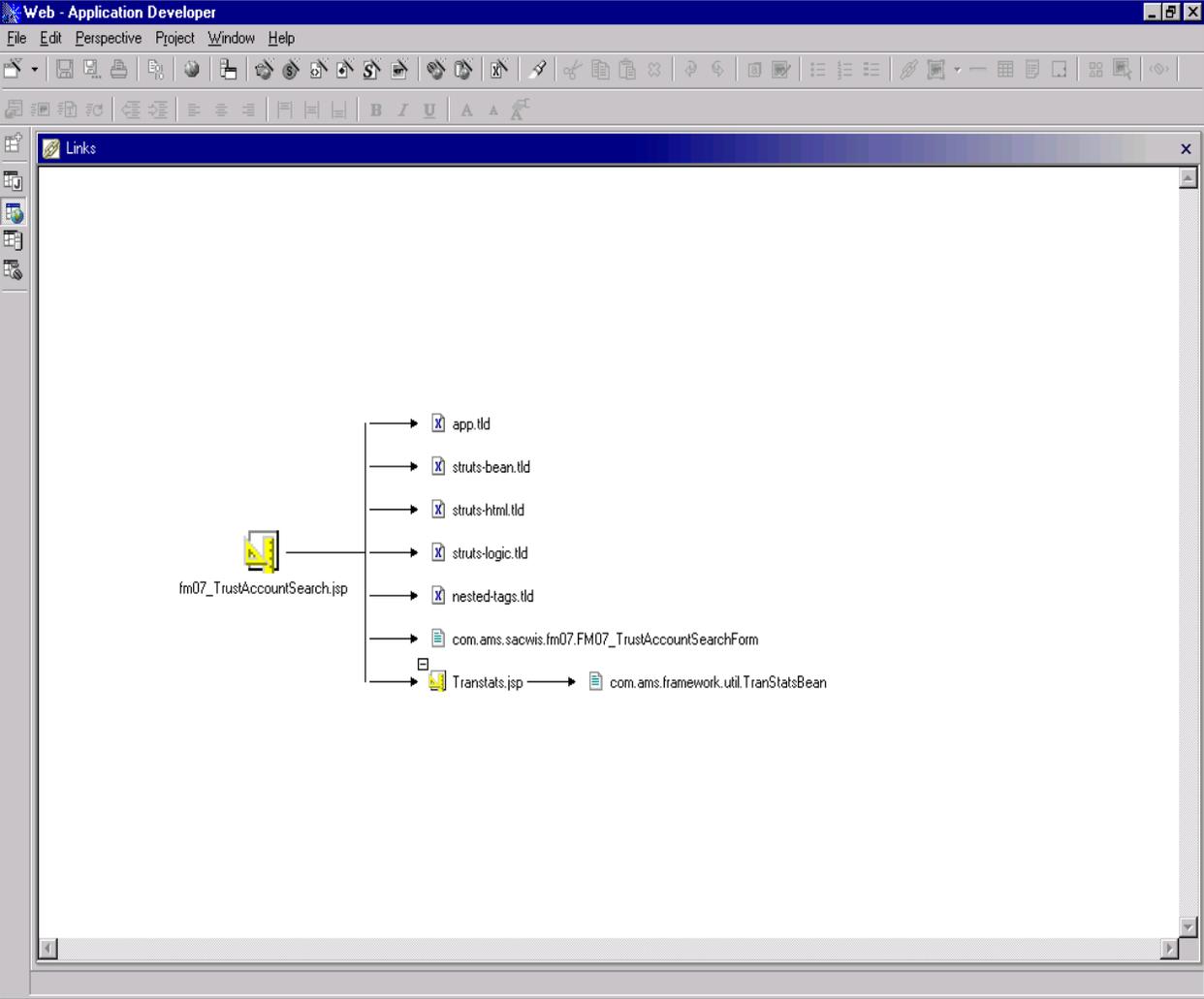
fm07_TrustAccountBenefit.jsp



fm07_TrustAccountLedger.jsp



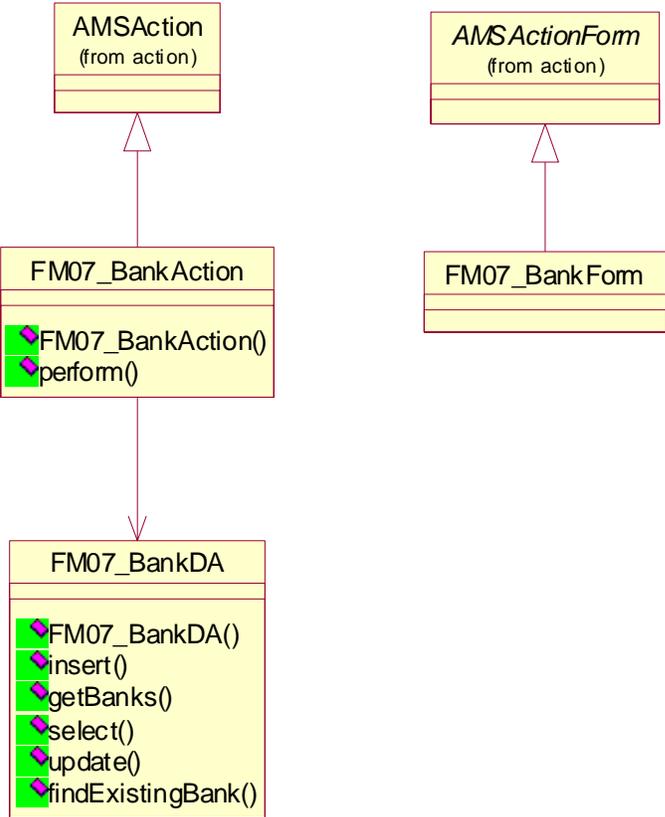
fm07_TrustAccountSearch.jsp



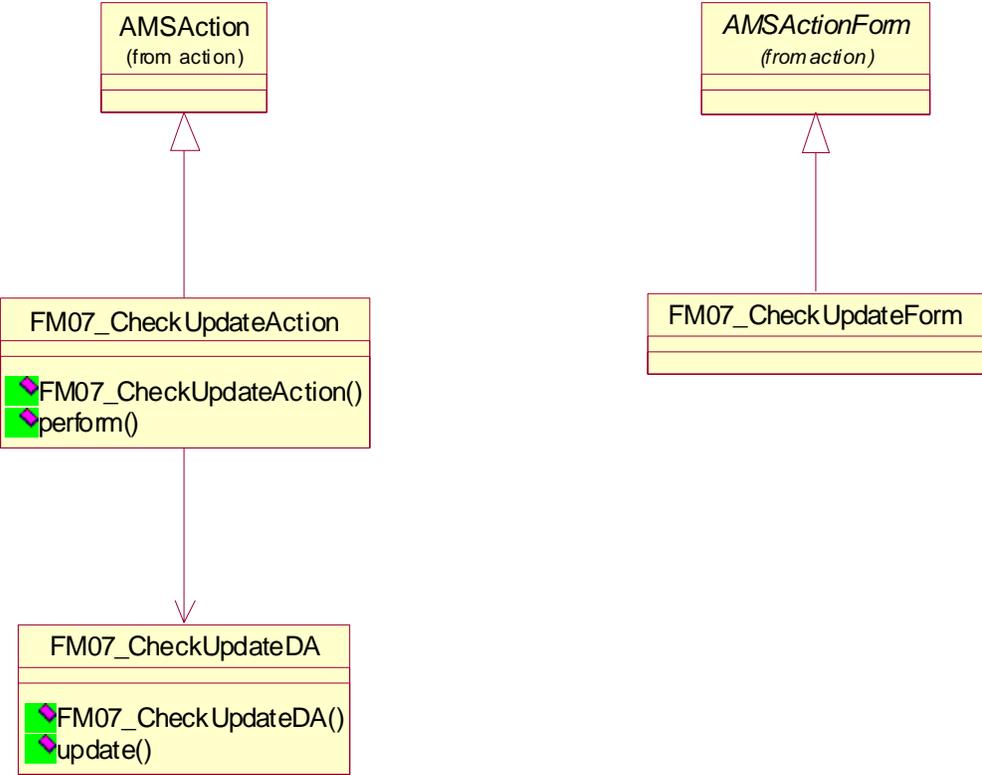
Java Components

Below is the object composition diagram for the CM01 topic. There are two distinct “families”. One that services all the Person Management transactions and another that services all the Address Management transactions.

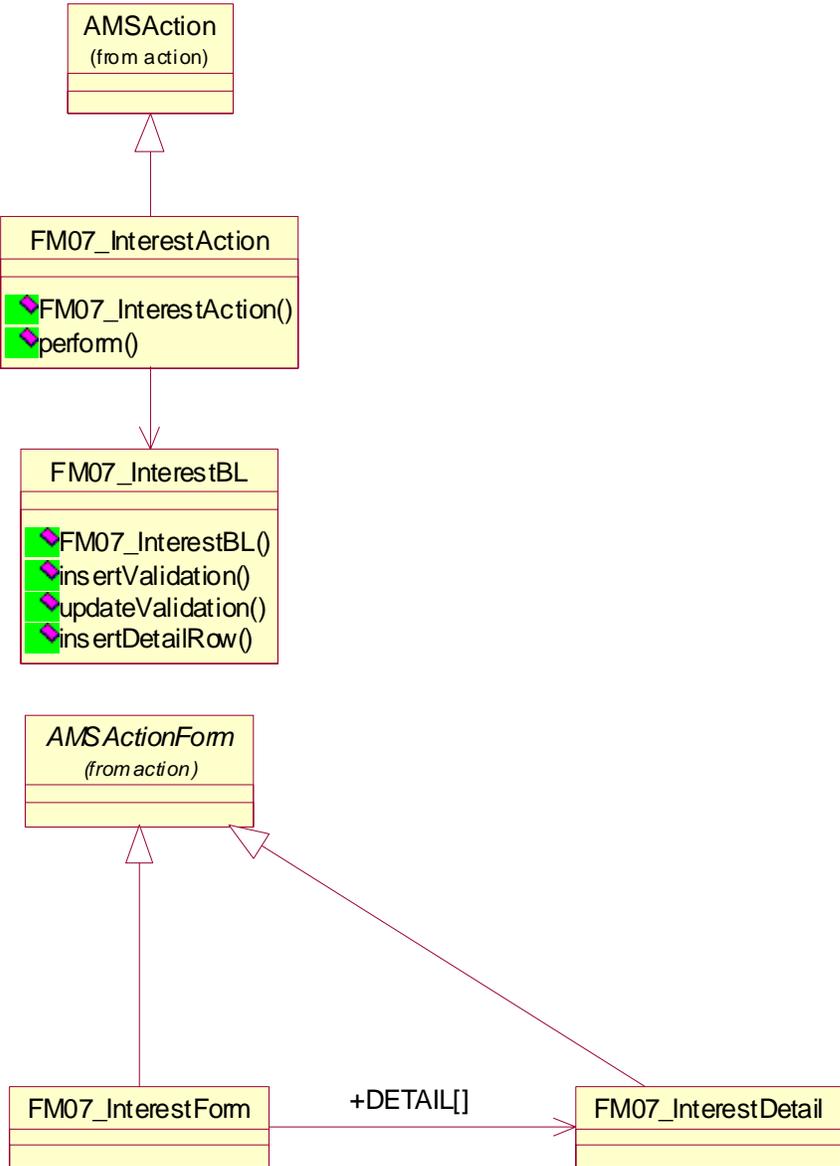
Main Classes involved in Bank transactions.



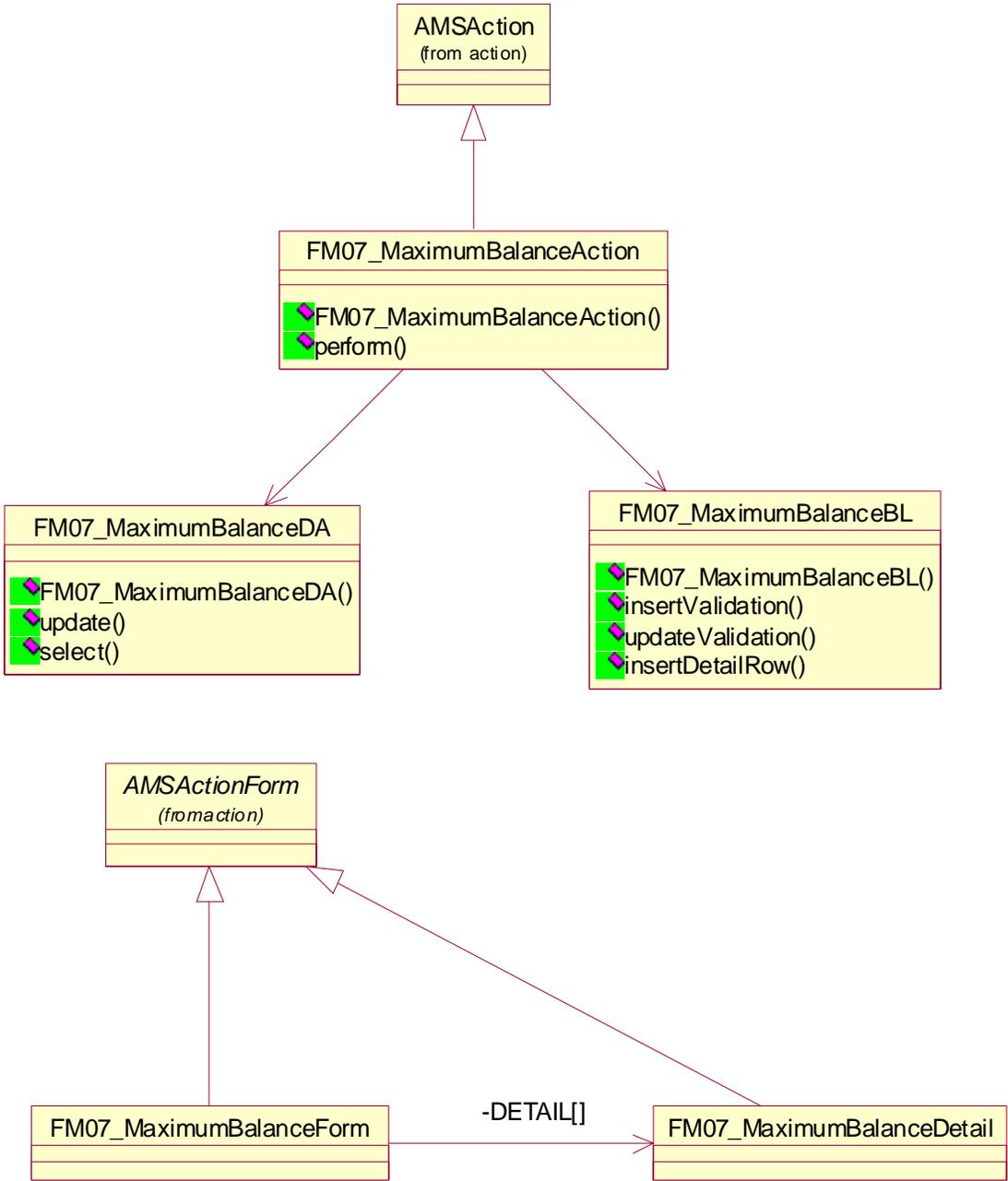
Main Classes involved in Check Update transactions.



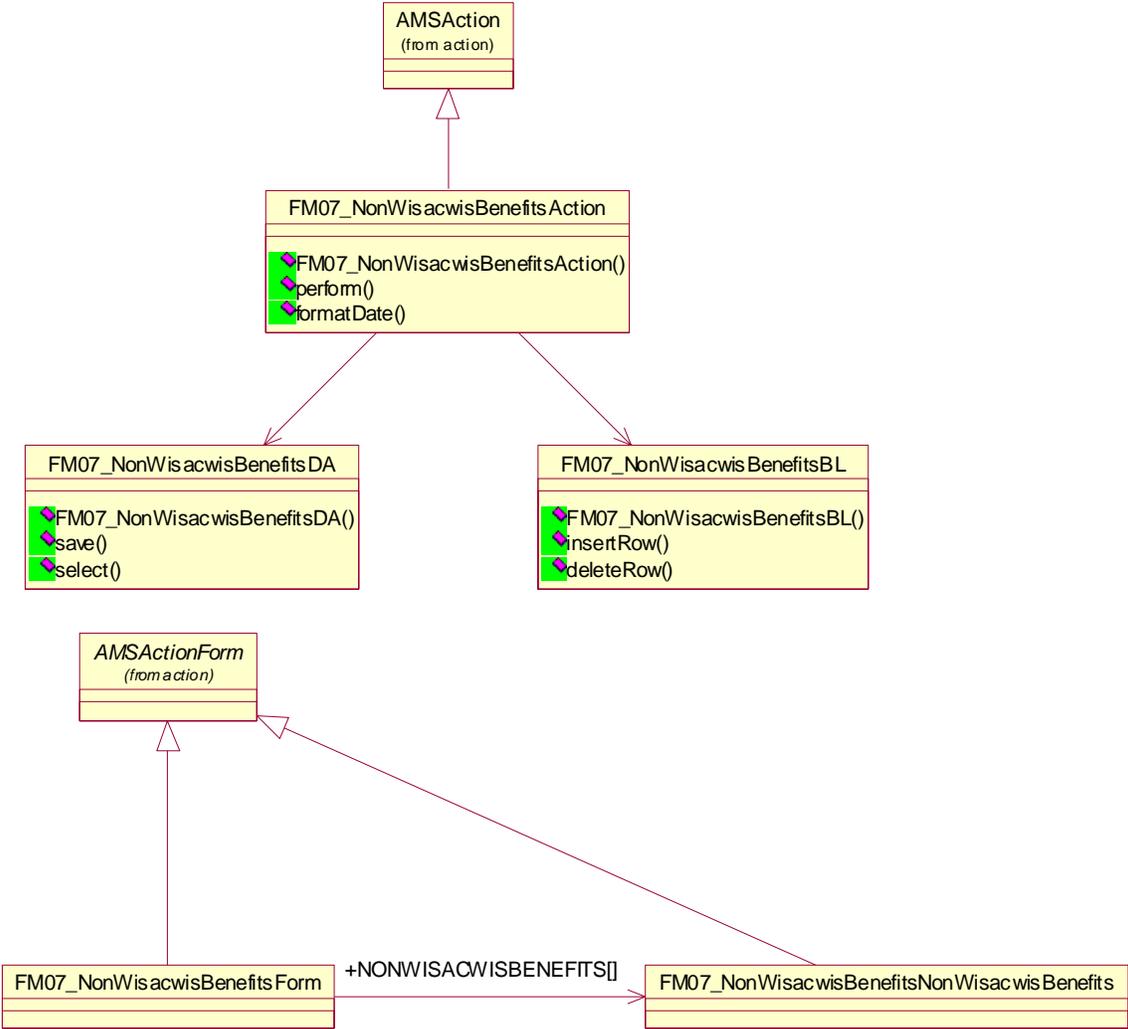
Main Classes involved in Interest transactions.



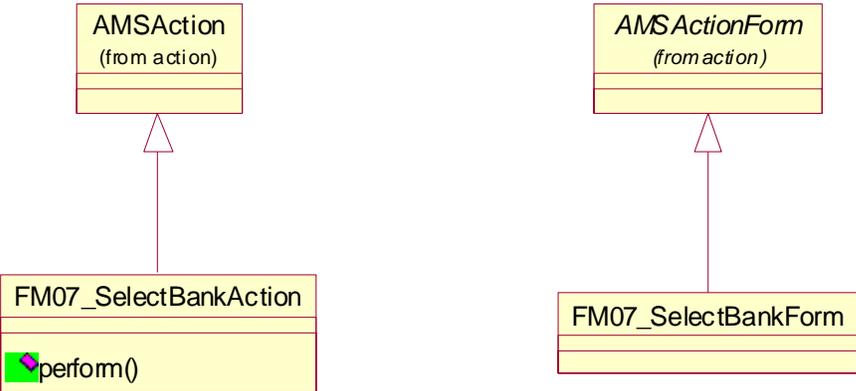
Main Classes involved in Maximum Balance transactions.



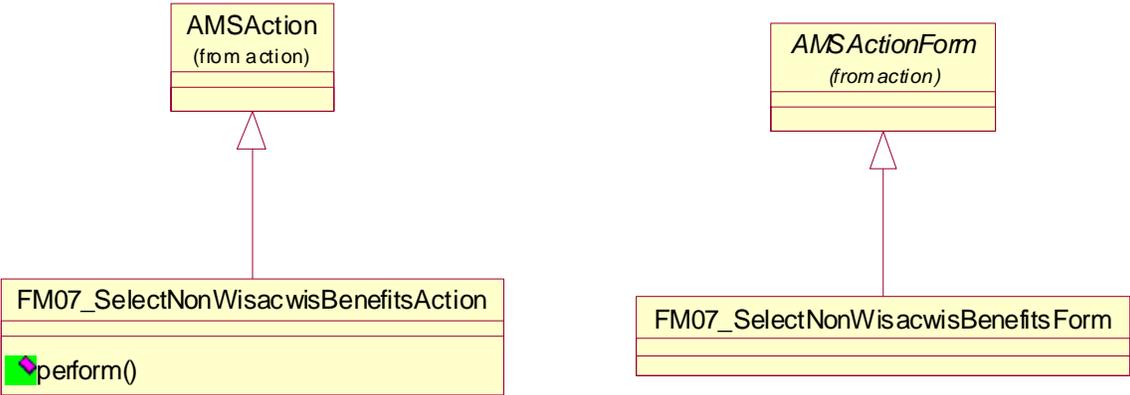
Main Classes involved in Non Wisacwis Benefits transactions.



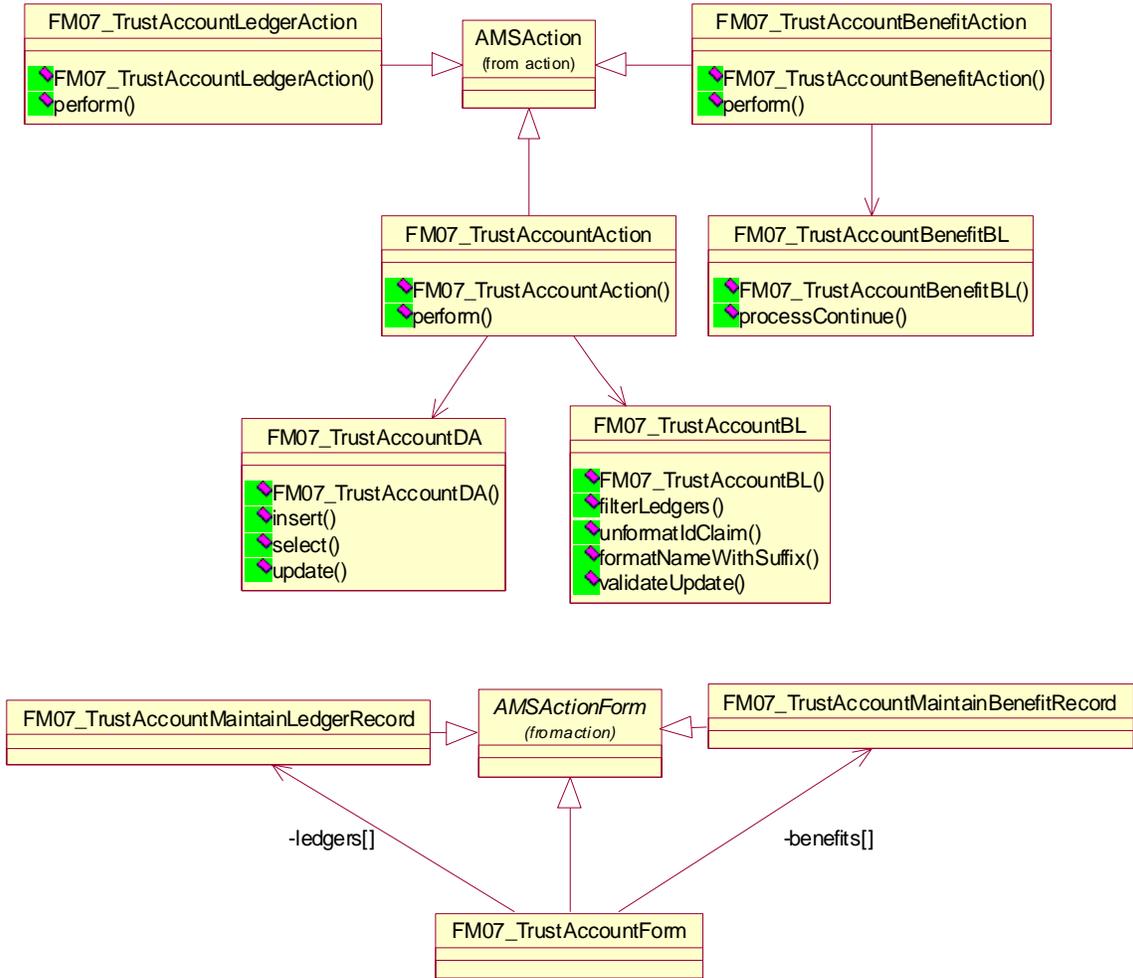
Main Classes involved in Select Bank transactions.



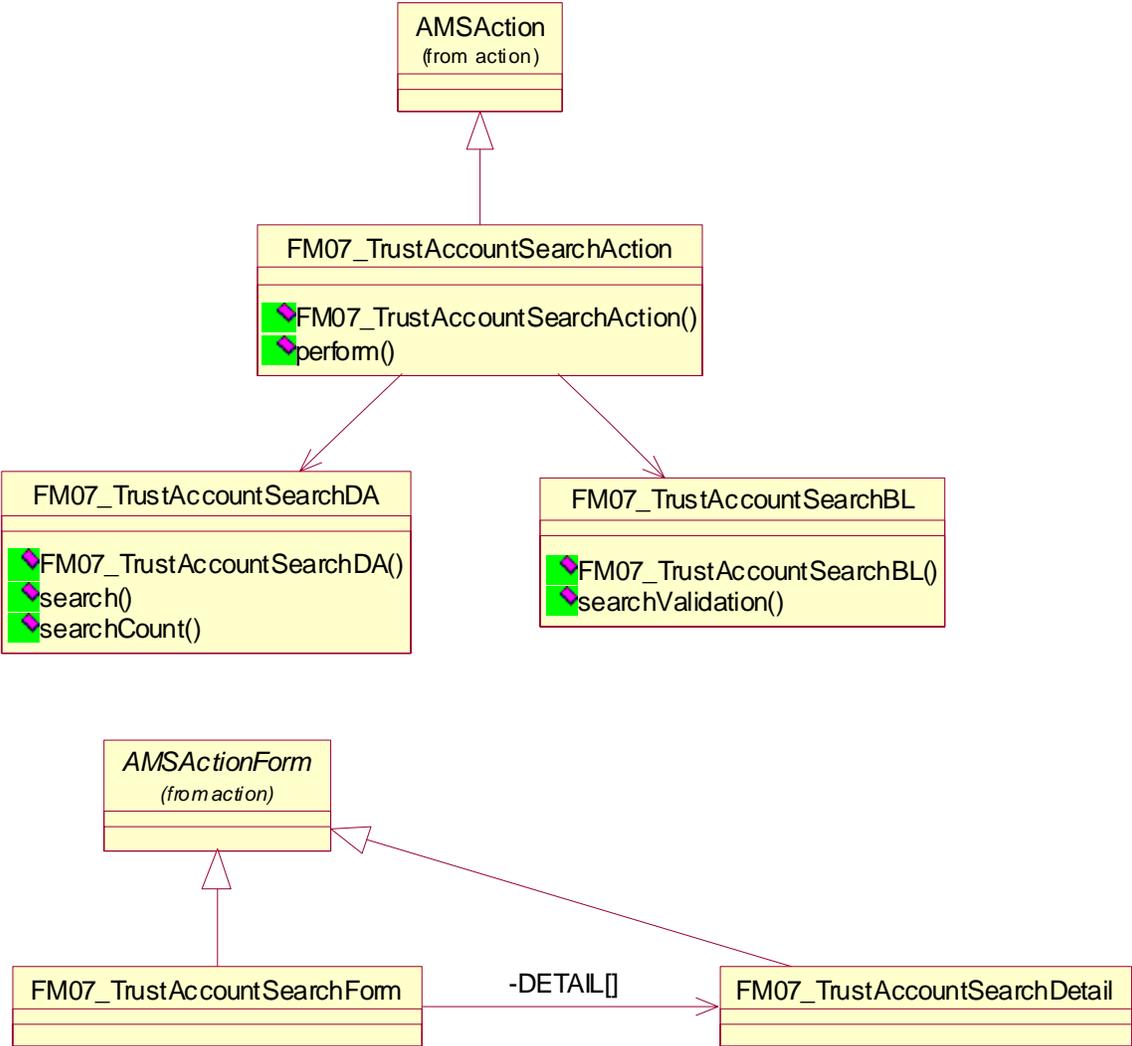
Main Classes involved in Select Non Wisacwis Benefits transactions.



Main Classes involved in Trust Account transactions.



Main Classes involved in Trust Account Search transactions.



Below is a brief description of each of the Java Classes:

| Type | Name | Comment |
|----------------|--------------------------------|--|
| Action Class | FM07_BankAction | Standard Action Class. |
| Data Access | FM07_BankDA | Standard Data Access Class. |
| Form | FM07_BankForm | Aggregate Data container for Bank information. |
| Action Class | FM07_CheckUpdateAction | Standard Action Class. |
| Data Access | FM07_CheckUpdateDA | Standard Data Access Class. |
| Form | FM07_CheckUpdateForm | Aggregate Data container for Check Update information. |
| Action Class | FM07_InterestAction | Standard Action Class. |
| Business Logic | FM07_InterestBL | Standard Business Logic Class contains Interest business rules. |
| Form | FM07_InterestForm | Aggregate Data container for Interest information. |
| Form | FM07_InterestDetail | Sub form of Interest Form. Data container for Interest Detail information |
| Action Class | FM07_MaximumBalanceAction | Standard Action Class. |
| Data Access | FM07_MaximumBalanceDA | Standard Data Access Class. |
| Business Logic | FM07_MaximumBalanceBL | Standard Business Logic Class contains Maximum Balance business rules. |
| Form | FM07_MaximumBalanceForm | Aggregate Data container for Maximum Balance Information |
| Form | FM07_MaximumBalanceDetail | Sub form of Maximum Balance Form. Data container for Maximum Balance Detail information. |
| Action Class | FM07_NonWisacwisBenefitsAction | Standard Action Class. |

Below is a brief description of each of the Java Classes:

| Type | Name | Comment |
|----------------|---|---|
| Data Access | FM07_NonWisacwisBenefitsDA | Standard Data Access Class. |
| Business Logic | FM07_NonWisacwisBenefitsBL | Standard Business Logic Class contains Non Wisacwis Benefits business rules |
| Form | FM07_NonWisacwisBenefitsForm | Aggregate Data container for Non Wisacwis Benefits information. |
| Form | FM07_NonWisacwisBenefitsNonWisacwisBenefits | Sub form of Non Wisacwis Benefits Form. Data container for Non Wisacwis Benefits Detail information |
| Data Access | FM07_SelectBankAction | Standard Action Class. |
| Form | FM07_SelectBankForm | Aggregate Data container for Select Bank information. |
| Action Class | FM07_SelectNonWisacwisBenefitsAction | Standard Action Class. |
| Form | FM07_SelectNonWisacwisBenefitsForm | Aggregate Data container for Select Non Wisacwis Benefits information. |
| Action Class | FM07_TrustAccountLedgerAction | Standard Action Class. |
| Action Class | FM07_TrustAccountAction | Standard Action Class. |
| Data Access | FM07_TrustAccountDA | Standard Data Access Class. |
| Business Logic | FM07_TrustAccountBL | Standard Business Logic Class contains Trust Account business rules. |
| Action Class | FM07_TrustAccountBenefitsAction | Standard Action Class. |
| Business Logic | FM07_TrustAccountBenefitsBL | Standard Business Logic Class contains Trust Account Benefits business rules. |

| Type | Name | Comment |
|----------------|---|---|
| Form | FM07_TrustAccountForm | Aggregate Data container for Trust Account information. |
| Form | FM07_TrustAccountMaintainBenefitsRecord | Sub form of Trust Account Form. Data container for Trust Account Maintain Benefits Record information |
| Form | FM07_TrustAccountMaintainLedgerRecord | Sub form of Trust Account Form. Data container for Trust Account Maintain Ledger Record information. |
| Action Class | FM07_TrustAccountSearchAction | Standard Action Class. |
| Data Access | FM07_TrustAccountSearchDA | Standard Data Access Class. |
| Business Logic | FM07_TrustAccountSearchBL | Standard Business Logic Class contains Trust Account Search business rules. |
| Form | FM07_TrustAccountSearchForm | Aggregate Data container for Trust Account Search information. |
| Form | FM07_TrustAccountSearchDetail | Sub form of Trust Account Search Form. Data container for Trust Account Search Detail information. |